

版权注意事项：1、书籍版权归著者和出版社所有；
2、本PDF仅用于个人获取知识，进行私底下知识交流；
3、PDF获得者不得在互联网以任何目的进行传播；
如有需要，请尽量购买正版实体书！支持书籍作者！！

付哲◎编著

海量运维监控 系统规划与部署

(基于Linux+Nagios+Centreon+NagVis等)

- ◎ 企业级IT运维监控系统变迁解析
- ◎ 资深运维监控专家的理论思维
- ◎ 知名互联网企业的运维监控实践



运用“工匠精神”精雕细琢属于自己的海量IT运维监控系统。

清华大学出版社



付哲

系统分析师、民航局机场工程专业高级工程师，现任首都机场信息技术部主管工程师。全面负责首都机场集成系统、安检信息系统、自动化运维监控系统的研发与运营工作，对企业级海量IT运维支撑、自动化运维平台规划、性能优化、成本控制、平台搭建、质量效率、系统高可用性管理、业务连续性治理等有丰富的经验积累。

付哲◎编著

海量运维监控

系统规划与部署

(基于Linux+Nagios+Centreon+NagVis部署)

清华大学出版社
北京

内 容 简 介

今天,互联网大潮催生了众多卓越企业,基于云计算和移动互联网的各类应用以及服务已经融入了大众生活。与传统企业相比,互联网企业的用户及业务规模很容易达到海量级别,在为用户提供优质业务服务的同时,企业内部对IT运维管理的质量水准也日益提出高标准和严要求,而IT运维管理的核心业务之一,IT运维监控工作就变得愈加重要。本书针对海量IT系统的特点,不仅提倡IT运维监控系统要基于Nagios和Centreon等开源系统量身定做,采取开源监控技术与企业IT服务和运维管理流程相结合的技术路线,而且从开源监控系统的规划、管理、流程/规范、系统/平台、监控、告警、安全、部署实施、优化、考核、持续优化和提升等诸多方面来与大家详细分享体会。

本书共分14章,涵盖的内容主要包括:带领读者深度了解Nagios和Centreon如何在Linux系统上部署,以及如何与NagVis进行集成;从专家角度介绍如何管理Centreon、Nagios和NagVis,以及如何运用相关技巧优化这套组件以提升监控系统效率;运用大量脚本样例和截图,手把手帮助读者解决在构建开源监控系统中遇到的各类实际问题;利用NagVis和RRDTool集成开源监控系统的视图功能;按部就班地协助用户定制化实现既符合ITIL最佳实践,又符合企业自身特点的企业级IT运维监控系统。

本书适合在互联网企业以及传统企业内部,那些想了解、学习、规划以及快速构建开源IT运维监控系统的人员阅读,可以作为学习Nagios和Centreon的工具书,也适合有一定基础,想更深入学习Centreon的读者,通过大量的案例,让读者真正理解Linux、Nagios、Centreon和NagVis这一套犀利武器,为海量IT运维监控工作保驾护航。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

海量运维监控系统规划与部署(基于Linux+Nagios+Centreon+NagVis等)/付哲编著.-北京:清华大学出版社,2015

ISBN 978-7-302-40953-3

I. ①海… II. ①付… III. ①计算机监控系统 IV. ①TP277

中国版本图书馆CIP数据核字(2015)第166274号

责任编辑:栾大成

封面设计:杨玉芳

责任校对:徐俊伟

责任印制:李红英

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦A座 邮 编: 100084

社总机: 010-62770175 邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者: 北京鑫丰华彩印有限公司

装 订 者: 三河市漂源装订厂

经 销: 全国新华书店

开 本: 188mm×260mm 印 张: 22.5 插 页: 1 字 数: 565千字

版 次: 2015年11月第1版 印 次: 2015年11月第1次印刷

印 数: 1~3500

定 价: 59.00元

产品编号: 063810-01

序 言

认识付哲是在 2007 年。在 2008 年，随着 T3 航站楼主体建筑的落成，首都机场迎来了关键的核心信息系统切换工作，而且要求一次性切换成功，不许失败。作为身负重要使命的首都机场信息团队中的一员，付哲完成了多个重要信息接口的开发工作，由此我也知道了他的绰号——Futeface，也了解到他是我在北京航空航天大学的校友。2014 年，首都机场的信息系统监控平台顺利投产，听闻主持该项目工作的付哲又有了新的著作，即这本即将由清华大学出版社出版的书。

付哲的这本新书便是能够指导读者成为拥有多维视角的运维监控专家的指南图书。这本书包含了构建监控平台的最佳实践、注重实用和快速上手。本书对 Nagios、Centreon 以及 NagVis 的安装步骤和运作原理进行了介绍，包含了实施开源 IT 运维监控项目的实用指南。作者编写这本书虽然以帮助运维工程师和运维软件架构师为主，但书中包含的内容依然与 ITIL 流程中大多数技术角色相关。作者在书籍中避免了以官方在线文档为内容的乏味介绍，而是从基础出发，提供了运用开源软件进行监控系统集成的细节信息，并结合自身使用这组套件的经验介绍了一些高级主题，带领读者对 Nagios 的软件生态系统进行系统化的思考。

在本书介绍的这套开源软件组合最佳实践方案中，作者明确了在监控系统的建设过程中，对被监控的业务和技术的深刻理解非常重要，实施过程应当是经过事先规划，深思熟虑，而非东拼西凑的。与常见书籍中侧重于安装步骤的按部就班执行以及配置参数的堆砌有所不同，本书着重介绍的是开源软件的深层次技术细节，在配置参数和技术选型方案上反复推敲，从而形成完善的开源运维监控平台技术方案。

尽管本书前面向读者介绍了大量的与系统实施有关的事前规划、架构设计、业务知识、安装步骤、配置参数等等知识，但作者依旧避免对被监控的大量系统作出假设。事实上，Nagios 自身无任何监控功能，设计它的目的是对监控检测的调度（Schedule），并根据检测结果进行相应的通知（Notify）。Nagios 将实际的监控功能委托给能够返回状态文本的插件，通过这种方式，它可以避免对一体化的 Agent 产生依赖，最大限度降低宿主的负担，从而符合 Doug McIlroy 所倡导的 Unix 的设计哲学：

“编写专一并且专注的程序（只做一件事，并把它做好）。编写能够协作的程序（程序间能够相互调用）。编写能够对文本流进行处理的程序，因为这是一种通用的接口。”

事实上，IT 运维监控工作是一项重要而平凡的工作，IT 运维监控人员都是在幕后默默奉献的无声英雄。如何使工作平凡而不平庸，在平凡中透出精致，让 IT 运维监控的人生充满智慧与成就感，是作者希望通过此书用技术手段表述的意境。

高利佳

北京首都国际机场股份有限公司 正职级常务副总经理

前言

在大型企业，尤其是互联网企业内部，在向公众提供各类业务服务的同时，背后的 IT 服务支撑、运维的角色越来越重要。企业的很多产品从无到有，从小到大，持续经历着经年累月的系统迭代、运行维护以及应急救援，在这些或大或小项目的生命周期中，固然离不开规划、研发、测试、部署等角色的全程参与和配合，但运维在上线前的架构、系统、网络、资源规划、部署及上线后的质量、效率、成本管理方面更是发挥了不可替代的作用。

在日渐汹涌的互联网浪潮和海量数据面前，无论是传统企业还是新生的互联网企业，普遍面临着产品的快速迭代和用户对于服务中断的零容忍。运维人员手中缺乏灵活高效的工具来支持 IT 运维管理和业务的深度融合，现有的诸多监控平台仅仅支持监控指标的堆砌，很少能够灵活反映业务关键节点的健康度，当企业的 IT 业务规模、访问量和运行环境发生变化时，传统 IT 运维监控平台的反应就稍显笨拙。

另一方面，自动化在运维管理中的作用越来越大，传统的人工检查和巡检方式已经无法满足运维规模扩大的需求，需要从流程化、标准化、自动化去构建能够支持海量数据的 IT 运维监控体系，提前预知故障。

幸运的是，面对用户对于性能提升或者业务优化的需求，产品研发人员和运维人员之间的界限愈加模糊。不仅优秀的技术架构师、项目管理者、研发工程师、测试工程师等角色都在深入了解运维监控工作，而且各类具备开发背景的运维人员同样运用自身的优势，在不同角色间主动参与、换位思考、跨界工作，不断推动运维监控工具的标准化、流程化、自动化。在此背景下，涌现出了众多杰出的开源 IT 运维监控工具，形成了成熟的社区以及生态环境，这其中，就有以运行在 Linux 操作系统上的佼佼者 Nagios、Centreon 和 NagVis。

IT 运维的核心工作是运行监控，本书即围绕此主题展开。本书的名字叫《海量运维监控系统规划与部署——基于 Linux+Nagios+Centreon+Nagvis 等》。海量一般适用于大型企业，其 IT 运维的特点是系统遵循行业标准，由业务流程驱动，具备大规模的架构、网络、系统、应用，并且从企业形象和安全的角度出发，对 IT 运维监控工作的数量和质量要求均高于普通应用场景。“基于 Linux+Nagios+Centreon+NagVis”是选择并介绍如何管理这套开源监控系统，提升其运行的质量、效率、满足企业定制需求并降低成本。本书详细讲述了以上两者结合的方法论，重申了 IT 运维监控角色在 IT 服务中的核心地位，为如何高效便利地利用开源系统实施 IT 运维监控工作指明了方向。

本书从管理、技术双视角对这套开源监控系统组合的功能进行了详细介绍。

从面向服务的运维管理与业务连续性治理角度出发，本书介绍了如何选择并使用最新的开源技术，搭建兼具低成本和高效益、高安全等级、符合 ITIL 最佳实践的可扩展基础监控框架，以及如何延伸扩展以适应各类规模的企业 IT 系统。

以自动化运维视角出发，重点讲述了 Linux、Nagios、Centreon 和 NagVis 这 4 类开源系统的安装配置，对自动化功能、监报告警、性能调优、协议、管理、优化，结合 Centreon 实现自动化配置管理等内容进行了全方位的深入剖析。从基础着手，由浅入深地重点讲解 Centreon 监控系统这个开源软件。从最简单的安装配置，到复杂的高级使用，详细讲解了监控项配置管理、系统管理、性能调优、架构设计，提供了大量的案例，对即将构建

Nagios+Centreon 监控系统或者已经在使用 Nagios 的用户具有非常高的参考价值。

本书进一步印证了企业系统的安全性和开源系统的灵活性并不冲突，而是存在深度融合的可能。成熟的、经过众多技术人员和使用者验证的、社区活跃的开源系统并非想象中的不安全，不仅能够被大规模运用在互联网行业，同样因其灵活可控且经过实践验证而适用于企业级场景。而开源的精神就是分享，让更多人受益的同时，自身的水准也在持续提升。经常看到很多集成商和 IT 运维人员都在做监控平台，但这些监控系统的功能事实上惊人相似，重复劳动意义不大，闭门造车更无济于事。开源的精神就是一个人共享出来，大家一起来使用、完善，达到众人拾柴火焰高的效果。对整个行业来讲，投入成本都会降低，对个体来讲也是资源的整合。如果形成良性循环，行业的生态环境将会有很大程度的改善。本书作者在对安全性有极高要求的民航业工作，同时热衷于开源技术，同样也愿意为开源贡献一分微薄之力，希望更多的人能支持开源、参考开源。

勘误和支持

尽管作者做了很多努力，尽力使本书不出现重大疏漏，但出于专业积累和沉淀等原因，本书仍然会有瑕疵。诚愿各位读者和专家发现后及时与作者本人联系，在此对支持本书的读者表示最真挚的谢意。如果您有更多的宝贵意见，欢迎发送邮件至邮箱 cauc@163.com，期待能够得到你们的真挚反馈。

另外，还可以添加专业运维监控公众号，获得最新动向。



致谢

首先要感谢 Ethan Galstad 大神，是他创立了 Nagios 及社区，同时也要感谢提供 Nagios 优秀插件的所有作者以及 Centreon 的作者，开源的精神与力量在他们身上体现得淋漓尽致。

感谢北京首都国际机场股份有限公司正职级常务副总经理高利佳、信息技术部总经理熊英、商业开发部总经理肖挺莉，是她们给予我第一份工作，也为我此后的成长提供了非常多的指导。感谢北京航空航天大学计算机系姚淑珍教授在校期间给予我的专业指导。感谢北京首都国际机场股份有限公司提供了这么优秀的平台，让我有机会可以尽情施展才能，体现个人价值。感谢首都机场信息技术部的张喆、向红艳、李敏乐、李颖等优秀同事以及 SOCC 的所有兄弟姐妹在工作中给予的帮助、指导与支持，让我可以在新的环境继续突破自我，实现自我价值。感谢读研究生期间的同学邵海刚，在他的影响下才促成了这本书的写作与出版。

感谢清华大学出版社的编辑栾大成，在这半年多时间中始终富有激情地支持我的写作，他的鼓励和帮助引导我能顺利完成全部书稿。

最后感谢我的爱人韩杨同学，没有她就没有我们幸福的小家。感谢她支持我做的所有决定，没有她背后默默的支持与包容，也没有我今天的成就，更不会有这本书。我想对她说：“谢谢你！有你真好”。

付哲

目 录

第 1 章 企业级 IT 监控系统概述	1
1.1 什么是 IT 运维监控系统	2
1.2 开源监控软件之崛起——Linux、Nagios、Centreon 和 NagVis	3
1.3 Nagios 简介	5
1.3.1 云计算和海量运维监控的最佳选择	6
1.3.2 Nagios 的主机检测与服务检测	7
1.3.3 监控信息的提供者	7
1.3.4 及时的通知机制	8
1.3.5 从外部系统接收信息	9
1.3.6 Nagios 与 Linux 的关系	9
1.4 Centreon 简介	10
1.4.1 Centreon 引擎	11
1.4.2 为什么要有 Centreon 引擎	11
1.5 NagVis 简介	12
1.6 为什么要基于开源软件构建 IT 运维监控系统?	13
第 2 章 企业级 IT 运维监控系统的构建——从源代码到企业级系统	17
2.1 可供选择的操作系统	18
2.1.1 选用 Red Hat Enterprise Linux 作为操作系统	19
2.1.2 选择部署方式	19
2.2 服务器安装规划	19
2.2.1 服务器参数规划	20
2.2.2 服务器存储规划	20
2.3 Linux 的逻辑卷 (LVM) 管理机制	21
2.3.1 为什么要使用 LVM	21
2.3.2 LVM 基本概念	21
2.3.3 操作系统分区划分样例	23
第 3 章 配置 VMWARE 虚拟机	25
3.1 新建虚拟机向导	26
3.2 VMware 的联网模式简介	28
3.2.1 虚拟网络设备	28
3.2.2 虚拟机联网方式之桥接模式 (bridged networking)	29
3.2.3 虚拟机联网方式之网络地址转换 (network address translation, NAT) 模式	30
3.2.4 虚拟机联网方式之仅主机 (host-only networking) 模式	31

3.2.5	关于虚拟机联网方式中的 DHCP 服务	32
3.2.6	选择 Nagios 虚拟服务器的联网方式	33
3.3	完成虚拟机创建向导并查看配置清单	33
第 4 章	为虚拟机安装 RHEL 操作系统	35
4.1	引导菜单	36
4.2	操作系统安装欢迎界面 (语言及键盘布局)	36
4.3	存储设备选择	38
4.4	主机名与网络设置	39
4.5	时区选择	41
4.6	磁盘分区设置	42
4.7	划分文件系统	43
4.8	安装操作系统软件	45
4.8.1	格式化虚拟机硬盘	45
4.8.2	选择操作系统安装类型	48
4.8.3	安装操作系统	50
4.8.4	操作系统初始化配置	51
4.8.5	创建操作系统账户	52
4.8.6	设置操作系统时间	52
4.8.7	设置 Kdump	54
4.8.8	操作系统网络配置	55
4.8.9	yum 源配置	55
第 5 章	Nagios 的安装	59
5.1	Nagios 安装前的准备工作	60
5.2	创建 Nagios 用户和组	61
5.3	编译并安装 Nagios	62
5.4	安装 Nagios 插件	66
5.5	配置 Nagios 的 Web 用户界面	67
5.6	SELinux	69
5.7	访问用户认证与授权	70
第 6 章	NDOUtils 安装	75
6.1	配置并编译 NDOUtils	76
6.2	拷贝编译后的文件至运行目录	77
6.3	检查 MySQL 的配置	79
6.4	创建 NDOUtils 数据库表	80
6.5	配置 NDOUtils	86
6.6	添加 ndo2db 为系统服务	88

第 7 章 Centreon 的安装与配置	93
7.1 什么是监控以及如何监控	94
7.1.1 监控已经不再局限于基础设施	94
7.1.2 基础设施监控	94
7.1.3 应用程序监控	95
7.1.4 SLA 监控	95
7.1.5 业务活动监控	96
7.2 究竟什么是运维监控	96
7.2.1 运维监控的原则	96
7.2.2 主动监控模式	97
7.2.3 被动监控模式	98
7.3 SNMP	98
7.4 Centreon——不仅仅是包装后的 Nagios	99
7.4.1 MERETHIS 公司简介	99
7.4.2 Centreon 的功能	100
7.5 Centreon 的架构	102
7.5.1 系统组件	102
7.5.2 数据存储	103
7.5.3 检测命令	104
7.5.4 调度进程	105
7.5.5 其他兼容 Centreon 的调度引擎	106
7.5.6 代理进程	106
7.6 后台服务和定时任务	107
7.6.1 centcore 服务	108
7.6.2 centstorage 服务	110
7.6.3 定时任务	110
7.7 系统架构——简洁及分布式	112
7.8 捕获 SNNP trap 告警信息	115
第 8 章 安装 Centreon	117
8.1 安装前提	118
8.2 安装 Centreon 监控系统中央服务器	120
8.2.1 系统软件需求	120
8.2.2 部署 Centreon 监控系统	127
8.3 安装后配置	143
8.4 Centreon 的 Web 用户界面	149
8.5 Centreon 的语言设置	150
8.6 Centreon 的数据库连接配置	151
8.7 通过 Centreon 激活 Nagios 监控	152

8.8 安装过程中的问题解决	155
8.8.1 Export 时显示 sudo 相关错误	155
8.8.2 在/var/log/messages 中出现 Warning: queue send error 错误	157
第9章 Centreon 的管理	159
9.1 Centreon 的调度进程和代理进程	160
9.2 Centreon 对于 Nagios 调度进程的管理	160
9.2.1 Files 选项卡	162
9.2.2 Check Options 选项卡	163
9.2.3 Log Options 选项卡	165
9.2.4 Data 选项卡	167
9.2.5 Tuning 选项卡	168
9.2.6 Admin 选项卡	169
9.2.7 Debug 选项卡	170
9.3 Centreon 对于 NDOUtils 代理进程的管理	171
9.3.1 General 选项卡	172
9.3.2 Database 选项卡	172
9.3.3 Retention 选项卡	173
9.4 Centreon 对于 ndomod 的管理	173
9.5 Centreon 的实时监控	175
9.5.1 主机和主机组	175
9.5.2 服务、服务组和元服务	176
9.5.3 硬状态和软状态	177
9.5.4 状态波动与状态特殊震荡	178
第10章 Centreon 的实时监控	179
10.1 专注于实时监控的 Centreon	180
10.2 Centreon 的通用监控	182
10.3 状态总揽视图	183
10.4 全局健康视图	184
10.5 主机的实时监控	185
10.6 主机的详细信息视图	186
10.7 服务的实时监控	191
10.8 在实时监控界面中进行监控项相关操作	195
10.8.1 主机和服务操作概述	195
10.8.2 处于告警状态下的主机或者服务进行确认	196
10.8.3 计划停机	198
10.8.4 添加备注	202
10.8.5 对于调度任务的直接控制	203

第 11 章 Centreon 的配置	207
11.1 Centreon 的监控对象模型	208
11.2 通用功能配置界面	208
11.3 Nagios 配置文件的生成与部署	212
11.4 宏、检测命令与检测插件	216
11.5 检测命令与检测插件	220
11.6 执行周期	224
11.7 主机模板和服务模板	226
11.7.1 模板和继承	226
11.7.2 继承规则	226
11.7.3 主机模板	227
11.8 主机和主机组	232
11.9 主机的配置界面	233
11.9.1 “通用配置”选项卡	234
11.9.2 “关系”选项卡	236
11.9.3 “数据处理”选项卡	237
11.9.4 “主机扩展信息”选项卡	239
11.10 主机组	239
11.11 服务	240
11.11.1 “服务配置”选项卡	241
11.11.2 “关系”选项卡	243
11.11.3 “数据处理”选项卡	243
11.12 元服务	244
11.13 被动监控模式和 SNMP trap (SNMP 陷阱)	247
11.14 通知	253
11.14.1 通知策略定义	253
11.14.2 为主机和配置通知策略	255
11.15 通知消息联系人、联系人组以及联系人模板	257
11.16 Commands 通知命令	260
11.17 Escalation-告警通知的升级	261
11.18 性能图形	264
11.18.1 相关定义	264
11.18.2 查看图形与进一步分析	265
11.18.3 配置性能图形相关属性	268
11.18.4 配置性能曲线相关属性	270
11.19 利用性能图形实现早期预警	273
11.20 报表	276

第 12 章	Centreon 的管理和优化	279
12.1	Centreon 的管理菜单	280
12.2	通用选项	280
12.2.1	Centreon 的通用选项界面	281
12.2.2	Centreon 的监控选项界面	283
12.3	CentStorage 的相关配置	284
12.3.1	性能数据的配置管理	285
12.3.2	度量和计量	286
12.3.3	监控性能指标的相关操作	287
12.4	访问控制列表 (ACL)	288
12.4.1	访问控制列表的配置与管理	289
12.4.2	访问组	293
12.5	调度进程的运行时统计信息	293
12.6	Centreon 监控平台的备份与恢复	296
12.6.1	系统备份	296
12.6.2	系统恢复	301
第 13 章	NagVis 的安装与配置	303
13.1	关于 NagVis	304
13.1.1	地图关系设定	304
13.1.2	NagVis 的地图	305
13.2	NagVis 的运作机制	306
13.3	NagVis 的安装	307
13.4	NagVis 的配置	314
13.4.1	配置 NagVis 的默认参数	316
13.4.2	配置 NagVis 的后台数据源	317
13.5	NagVis 地图介绍	319
13.6	NagVis 地图的配置管理	320
13.7	NagVis 中背景图片的管理	322
13.8	配置 NagVis 的监控地图	323
13.9	设置 NagVis 图标超链接	325
13.10	设置 NagVis 的 Web 界面为自动登录	327
第 14 章	构建企业级 IT 运维监控系统	331
14.1	IT 服务管理和 ITIL	332
14.2	IT 运维监控系统与 ITIL 的关系	332
14.2.1	ITIL 的产生与发展	332
14.2.2	ITIL 的管理框架简介	333
14.2.3	运用 ITIL 解决企业 IT 服务管理面临的问题	336

14.3 企业级 IT 运维监控系统的构建与实施	339
14.3.1 咨询与梳理步骤	339
14.3.2 互联网运维监控实践	342
14.3.3 提升监控及预警能力	342
14.3.4 监控及预警质量的持续改进	344

企业级 IT 监控系统概述

随着互联网大潮的迅猛发展,以及对于传统行业的不断渗透,国内企业的信息化发展也取得了前所未有的成就,无论是部署规模还是系统规模都变得庞大起来。伴随而来的企业信息化需求也呈现出多元化、多层次、异构化,使得 IT 基础框架和上层应用日益复杂。对于从事企业 IT 运维工作的管理人员和技术人员来讲,为了提升信息服务质量、确保信息安全,如何及时获得信息系统告警信息、迅速定位故障原因、快速高效地处理各类 IT 问题、降低故障平均故障响应时间等等,就成了亟待解决的问题和难点。

目前,很多企业的核心业务都已经完全信息化,有了业务稳定可靠、快速有效地开展,企业经常会运用多个信息系统进行消息传递和系统交互,从而加大了故障定位的时间和解决问题的难度。面对数量庞大且分布业务中,每一位负责运维 IT 系统管理人员在面对用户的投诉、指导的问题、同事们的紧张时,无不会殚精竭虑地思考如何能够快速准确地定位系统故障,及时采取有效手段使故障能够快速解决,业务能够及时恢复。如此一来,亟需开发一套适合企业自身特点的,能够统一管理和展现各种监控资源、实现集中告警,全面协助 IT 运维管理人员实时掌握系统整体运行状态,快速定位故障、准确故障时间的企业级海量 IT 系统监控系统就显得迫在眉睫了。

第 1 章

企业级 IT 监控系统概述

随着互联网大潮的迅猛来袭,以及对于传统行业的不断渗透,国内企业的信息化发展也取得了前所未有的成就,无论是部署规模还是运维规模都变得庞大起来。伴随而来的企业信息化需求逐步迈向多元化、层次化、异构化,使得 IT 基础框架和上层应用日益复杂。对于从事企业 IT 运维工作的管理人员和技术人员来讲,为了提升信息服务质量、确保信息安全,如何及时获得信息系统告警信息、迅速定位故障原因、快速高效地处理各类 IT 问题、降低故障率和故障响应时间等等,就成了亟待解决的问题和难点。

目前,很多企业的核心业务都已经完全信息化。为了确保业务稳定可靠、快速有效地开展,企业经常会运用多个信息系统进行消息传递和系统交互,从而加大了故障定位的时间和解决问题的难度。面对服务器宕机或者业务中断,每一位负责任的 IT 运维管理人员在面对用户的投诉、领导的问责、同事们的紧张时,无不在殚精竭虑地思考如何能够快速准确地定位系统故障,及时采取有效手段使故障能够快速解决,业务能够及时恢复。如此一来,研发并部署一套适合企业自身特点的,能够统一管理和展现各种监控资源,实现集中告警,全面协助 IT 运维管理人员实时掌握系统整体运行状态,快速定位故障,缩短处理时间的企业级海量 IT 运维监控系统就显得迫在眉睫了。

1.1 什么是 IT 运维监控系统

既然 IT 运维监控系统这么重要，那么究竟什么才是 IT 运维监控系统呢？

所谓 IT 运维监控系统，有如下两层含义——“监”指的是对其他服务器的检测、监视；“控”指的是对其他服务器的控制，掌控。IT 运维监控系统往往是一套独立的信息系统、或者是若干信息系统的集合，用以对其他信息系统进行问题检测，甚至能够实现对其他信息系统进行部分或者完全的远程控制。

例如，就服务器检测而言，监控系统能够周期性地连接到一个 HTTP 服务器上，检测其是否能够正常响应浏览器的请求。又例如，监控系统能够接收系统管理人员的指令，在被监控的服务器上执行一个脚本，完成某项控制类操作等等。

如果实施得当的话，一套好的 IT 运维监控系统可以成为各类信息技术人员最好的朋友。它能在信息系统出现灾难之前就提前告知系统管理员某些细微的故障症候，使管理人员能够未雨绸缪，及早采取措施避免系统发生不可修复的错误。它也能够记录系统某些规律性的行为，使管理人员借以梳理并总结出信息系统的普遍行为，规划出系统的运行负载和服务能力。IT 运维监控系统还能够协助信息安全工程师发觉系统运行中的异常信息，能够实现 IT 运行的可视化，以帮助企业高层及时掌控信息系统的实时状态。如果 IT 运维监控系统更加智能的话，它甚至在发现故障之后自行解决故障，而不用值班人员在发现故障后凌晨给系统管理员打电话惊醒对方的美梦。也就是说，好的 IT 运维监控系统能够给企业信息技术人员和管理人员注入正能量，使大家能够非常愉快地投入每天的工作，而不是充当救火队员时刻紧张地准备冲到第一线。

但往往理想很丰满，现实很骨感。很多时候，我们遇到的往往是糟糕的监控系统，它带给我们的只有种种的不快，例如如下场景，您是否似曾相识：

- 某些监控系统在遇到系统故障时，常常不报警或者总是报警，不是让管理人员挨上级批评，就是被频繁的报警短信或者电话逼疯。一般来说，前一种情况往往是由于监控系统长时间没有得到有效维护，继而导致无法发出有效报警引起的；而后一种情况则是由于监控项得不到合理调整而频频触发监控阈值引起的。
- 某些监控系统往往在被监控端部署庞大的客户端程序，长时间运行后产生各种各样的问题，例如消耗服务器资源、触发服务器过度负载、引发安全漏洞、产生庞大的网络流量等。
- 某些监控系统缺乏服务商良好的技术支持。随着监控项的增多，监控项报警的能力逐渐丧失，效率越来越低，或者服务商提供的服务费用较高，增大了企业的运营成本。
- 某些监控系统技术封闭，管理人员缺乏对该系统的全面了解，在出现报警故障等问题时无法寻找有效的技术支持，影响系统安全。
- 某些监控系统架构封闭，可扩展性较差，无法针对业务灵活地添加或者调整监控项。
- 某些监控系统不支持监控数据采集入库、数据展示、报表统计等功能，导致管理人员无法针对系统性能数据进行故障趋势分析和容量分析。

在当下国内的 IT 生态环境中，中小型企业 and 初创的互联网企业占据绝大多数，它们普遍有着和大型企业一样甚至更为复杂的 IT 基础设施和业务系统，却不能拿出和后者同样的预算来雇佣同样高水平的 24 小时 IT 监控专家，更无法短时间内出资购买昂贵的商业监控软件或者相应的技术服务，长期承受着大型商业监控系统软件提供商或多或少的忽视。与此同时，这些企业的核心业务又离不开 IT 技术的推动，更无法承受 IT 系统不可用带来的种种损失。如果能够存在一套物美价廉的监控系统，既能适应中小型企业多样架构的 IT 环境，又具备良好的扩展性和兼容性，无疑会受到这些企业的热烈欢迎。在此，我向大家隆重推荐一款开源 IT 运维监控系统软件组合——Linux、Nagios、Centreon 和 NagVis。从操作系统到监控软件，从配置管理工具到可视化监控视图管理工具，这组软件将能够满足中小型企业甚至大型企业多样化的 IT 监控需求。借助其高效可扩展的架构设计和智能灵活的监控插件，能够满足各类纷繁复杂的监控需求。一句话概括来说：只有您想不到的，没有它做不到的。

1.2 开源监控软件之崛起——Linux、Nagios、Centreon 和 NagVis

谈到开源监控软件，就不能不提到在业界众所周知的“四大”IT 运维监控软件提供商——BMC、CA、HP 和 IBM。根据 Gartner 的报告，这四大软件厂商在同领域解决方案中仍然占据着统治性的地位。但这并不意味着“四大”厂商可以高枕无忧了，根据同一份报告，它们同样面临着内部的互相竞争以及来自开源监控软件的竞争。例如，调查报告显示，有 29% 的受访者认为可以在自己企业内部部署开源监控软件，而且这个比率还在不断升高（Gartner 报告：“Challenges Loom for 'Big Four' IT Operations Vendors” April 20, 2005）。

“四大”公司的 IT 运维监控解决方案作为一种成熟的、企业级 IT 运维管理平台，其优异表现是我们有目共睹的。但纵使是最为强大的武器，如果没有一个好的指挥官和一支卓越的实施团队，不懂得如何发挥武器的强大战斗力，那也不可能取得太多辉煌的战果。在“四大”IT 运维监控系统部署和运行的一些实践中，就出现过各种各样的误区，其中有商务上的、有管理上的、更有技术上的原因，以至于将系统的部署以及后续运维带入了窘境，这种情况在 IT 运维管理年预算不高的中小型公司中很常见。

Nagios 是于 2002 年异军突起的一个轻量级的开源 IT 运维监控框架，它原来的名字叫 Netsaint，是出于监控网络设备的目的而开发的。在 2002 年问世之初，略显稚嫩的它面临着 What's up Gold、Big Brother、Host Monitor 等小型监控软件，以及其他一些检测主机是否在线，是否存活的简单监控工具的强有力竞争。在 1.x 的版本中，1.2 发行版就已经非常稳定了，自此以来 Nagios 逐渐赢得了用户的信任，反过来又给它的开发者——Ethan Galstad 以更强的信心投入到后续开发中去（<http://www.nagios.org/about/history>）。从最初的简陋个人工具到无所不能的监视利器，对于正面临重量级企业运维监控系统的高昂成本和维护压力的 IT 运维工程师和管理人员而言，Nagios 的出现为曾经阴霾的天空带来了灿烂的阳光。

作为开源家族中的重量一员，Nagios 在设计之初，只能运行在 Linux 操作系统上，如 Redhat、CentOS、Debian 和 Ubuntu 等主流 Linux 发行版本中，大都能够看到 Nagios（从版本 1.0 到 3.0）的发行包。值得一提的是，Nagios 在 Linux 的 32 位版本和 64 位版本中都工作得很好，因此操作系统版本位数并不是部署和运行 Nagios 的障碍。一般来说，Linux 操作系统系统安装完毕之后，需要安装一系列 Development 包，才能正常地编译、安装并运行 Nagios。

除了主流 Linux 操作系统之外，部分商业 Unix 操作系统，例如 AIX、Solaris，它们的高版本也都能够良好地运行 Nagios。但与安装后便已具备 Nagios 编译和运行环境的 Linux 系统不同的是，这些商业 Unix 系统必须手动安装了诸如 GCC、MySQL、Perl 等必须的编译工具和运行环境之后，才能和 Linux 操作系统一样，编译和运行 Nagios。

俗话说，智者千虑，必有一失，愚者千虑，必有一得。诚然，Nagios 作为出色的开源监控框架，其稳定性和安全性毋庸置疑。但是，众所周知，Nagios 是出了名的“难搞死”，其可用性和界面友好性一直是运维监控管理人员吐槽的对象。Nagios 基于 Web 的用户界面完全是基于 CGI 编写，由 C 语言直接生成 Html 代码，其风格仍然处在上个世纪，对于现在见惯了各种华丽界面的用户来讲，确实是风格落后。更让人难以接受的是，Nagios 的配置文件至今仍然基于文本，需要用 Linux 下的文本编辑器编辑管理。且 Nagios 的不同配置文件之间关联复杂，当 Nagios 启动的时候需要检测配置文件之间，以及配置文件内各配置项之间的关联是否合乎规范，否则就会报出校验失败的错误信息，导致无法启动。

作为 Nagios 的开发者和维护者，要保持 Nagios 作为一款监控框架的严谨，就需要在安全稳定和易用友好两者之间做出取舍。由于 Nagios 是一款用来监控生产系统核心服务器的监控软件，其稳定性和可靠性应该是首要考虑的因素。基于以上权衡，Nagios 的开发人员选择安全而忽视界面友好度也就可以理解了。在 Nagios 的发行版中，包含了一个简单的 CGI 用户界面，该界面向 Nagios 用户提供了简单的告警展示功能，但不包括任何配置文件管理、用户管理等后台配置管理功能。为了弥补这些缺陷，开源世界的各位大神们就努力开发了一系列的 Nagios 后台管理工具和前台展示界面，例如 Nagios V-Shell、NagiosQL、ICINGA 等，其中最著名的莫过于法国人开发的 Centreon (<http://www.centreon.com/>) 这一款软件。

Centreon 是一款 Nagios 的前端管理软件，拥有其他 Nagios 管理工具无法比拟的优点。Centreon 具备强大的模板管理工具，支持批量添加主机和服务，能够自动建立主机和服务之间的关联，采用了 AJAX 技术，能够实现 Web 界面的自动刷新、ACL 权限管理、日志管理、告警展示图形等功能。Nagios 通过 NDOUtils 插件将监控数据写入后台 MySQL 数据库中，而 Centreon 可读取这些监控数据并实时地展示各类告警信息。使用 Centreon 的 Web 界面可以轻松地对 Nagios 进行配置管理，相较于以编辑文本的方式管理 Nagios 而言，很大程度上减轻了系统管理员的负担。因此，完全可以使用 Centreon 和 Nagios 来轻松搭建企业级的分布式 IT 运维监控平台系统。

有了开源的 IT 运维监控框架 Nagios，以及开源的 Nagios 后台管理工具 Centreon，我们的企业级 IT 运维监控平台独缺一款监控大屏展示工具，这方面的佼佼者无疑是 NagVis (<http://www.nagvis.org/>)。作为 Nagios 的图形化展示插件，顾名思义，NagVis 即是 Nagios Visualization (Nagios 可视化) 的简称。NagVis 允许用户上传一张 PNG 格式的图像作为背景，将被监控的主机或者服务以监控图标形式摆放在背景地图上，以实时地显示这些被监控对象的状态。NagVis 采用了 AJAX 技术，用户通过浏览器就可以任意地将被监控对象图标摆放在背景的任何位置。NagVis 会根据对象的状态显示不同的图标：红色表示紧急状态 (CRITICAL)，黄色表示警告状态 (WARNING)，绿色表示正常状态 (OK)，以及一个灰色背景的问号表示未知状态 (UNKNOWN)。除了上述图标之外，NagVis 还允许用户自定义文本标签，允许用户在监控对象之间做连接线以标注对象之间的依赖关系。用户可以用这些丰富的监控图标、连线、标签和背景来实时展示企业级 IT 运行环境的各类细节信息，

并将这些信息投放在监控中心的大屏幕上供实时查看。

Nagios、Centreon 和 NagVis 这三剑客，再加上 Linux 集群环境和 MySQL 数据库，构成了一套一体化企业级 IT 运维监控平台，省时省力省成本，能够使您的 IT 运维工作如虎添翼。

1.3 Nagios 简介

作为一款开源监控软件，与其他众多开源或者商业监控软件的功能类似，Nagios 可以持续监视并检测主机以及主机上众多应用程序的运行状态，并且探测到这些监控对象是否工作正常，一旦发生意外，即可及时发出告警信息。同时，与其他监控软件的不同之处在于，Nagios 并不会对这些主机或者应用发起主动检测，而是使用各种类型的插件——运行在被检测主机上的各类代码片段，来执行这些繁重的检测任务。作为一种框架式的监控软件，Nagios 能够在很大程度上减轻其所在监控服务器的负载，是一种模块化和灵活化的架构。

与其他闭源的监控系统不同的是，Nagios 遵循了开源模式，其优势就是能够随需应变地进行扩展而不用担心任何不便。与企图大包大揽的商业监控软件不同，Nagios 可以出色地与其他众多的开源工具进行交互，直至众星捧月，独步江湖的地位。

下面讲一个典型的例子，在规模稍大的企业里很容易见到。

现在是星期一早上，一家公司海外分支机构的经理正在紧张忙碌着，一封重要的公司内部电邮仍未收到。根据以往经验，这位经理判断是邮件服务器出现了故障——度过一个周末，它和公司员工一样重返工作，由于不堪重负而死机了。可是经理检查邮件服务器后发现，无论是空荡荡的邮件发送队列，还是各种日志文件都显示这台邮件服务器运转正常，那么故障出在哪儿呢？

作为分支机构的负责人，这位经理也许会猜测总部电子邮件服务器掉线或者死机——对 Ping 命令无响应——可能是问题的根本原因，结果遭到总部 IT 工程师的坚决否认。因为总部的 IT 工程师尝试了 Ping 海外分支办公室的邮件服务器，结果是 Ping 不通，但总部的网络又运行正常，于是该总部的工程师认为故障出在分支机构的邮件服务器上。结果是分支机构的经理和总部的 IT 工程师陷入了僵持状态，谁都不肯承认是己方出现问题，故障查找仍在继续……

其实问题的根本原因是连接公司总部和海外分支的 VPN 线路出现故障，网络自动切换至了备份线路，而备份线路上没有配置总部和海外分支之间的邮件服务器路由（路由器的配置同步往往由人工执行，人工意味着出错的概率很大）——该线路由第三方的网络基础设施提供商负责维护。三方均未认识到这一点，结果导致了工作邮件的延迟，公司业务不可避免地受到了严重影响。

让我们将注意力转移到 Nagios 上来，如果能够让 Nagios 监控该公司总部和分支机构之间的 VPN 线路以及备份线路，并配置故障告警短信的话，一旦出现链路切换导致的网络中断，管理人员就会第一时间得到短信通知，从而有充裕的时间解决问题，就不至于会在周一的邮件收发高峰时刻面对可能存在多种故障原因的问题而胡乱医治了。

随着现代企业 IT 系统规模的日渐庞大，已没有哪个企业的 IT 运维部门能够承受日复一日的人工巡检工作。企业的网络范围和服务边界的不断扩展，IT 运维部门迫切希望在系统崩

溃或者服务不可用的第一时间得到通知，可以提前公关并安抚客户，而不是被动等待客户的投诉或者抱怨。而 Nagios，这一广受欢迎的开源世界明星，正好解决了此类问题，使得系统管理人员能够提前感知系统故障并施展必要的预防措施或者解决手段。

Nagios 能够在系统发生警告（Warning）或者紧急（Critical）状态时迅速地通知相关人员，而什么情况下会产生警告信息或者紧急信息，则由系统管理员提前指定。通常情况下，Nagios 提供一个 Web 形式的信息系统状态汇总，以绿色、黄色、红色分别标明某一系统处于正常、告警、紧急状态。此外，Nagios 还能够将告警信息通知给特定关联的系统管理人员，无论是通过电子邮件还是借助手机短信，系统管理员都会在第一时间得到系统状态的最新通知。

1.3.1 云计算和海量运维监控的最佳选择

随着云计算和大数据应用的飞速发展，IT 行业也将运力和负载逐渐转移到云服务上来，鉴于云计算和海量数据带来的高效灵活地分配资源、自动化部署、动态迁移等优秀特性，越来越多的企业开始自建或者租用云计算数据中心，构建自己的大数据平台。而云计算数据中心和海量数据平台必须保障位于虚拟机中各类操作系统和应用系统的正常运行，在虚拟机或者云计算基础硬件设备发生故障前能够及时发现并排除单点故障、控制服务依赖，就需要采取富有弹性的监控和管理框架软件，且具备秒级监控和分析决策能力，而 Nagios 正是达到这一目标的不二选择。

海量不仅指用户数量的庞大、用户数据的几何级膨胀，还指的是机房和集群环境的迅速扩展。对于那些拥有大规模通用计算平台的公司而言，其机房空间的容量和所容纳集群服务器的数量都在迅速扩张。运维、监控和管理短时间内如此大规模膨胀的集群，没有现成的例子可以参照，也没有成熟的模式可以遵循，运维团队的工作带来了巨大的挑战。面对这些挑战，唯有自主打造自定义的、可快速进化的监控工具，借助于灵活的插件机制，方能实现自动化运维监控和数据化的管理。

服务器数量的激增，分布式部署范围的扩大，愈发要求监控工具具备全局的监控和展示能力。传统的运维监控人员通常只要面对几十台或者上百台服务器，规模不会太大，而且由于提供服务的种类较为分散，单个群集的规模也很小，往往只有最多 2 个节点。对于此类环境而言，普通的商业监控系统或者简单规模的 Nagios 监控系统即可满足需求，无分布式部署要求。但在大规模分布式集群中，工作任务和工作性质明显不同，首先，运维人员面临的服务器动辄就是三五千台甚至上万台，量级大幅提升；其次，分布式操作系统提供存储、CPU 调度能力、内存使用、网络等功能，是基本资源的包装整合，从逻辑上看，相当于一台计算机；最后，基于分布式系统开发的应用相当于一个分布式数据仓库和大数据平台，用户可以在上面做 ETL 处理、SQL 查询、数据导入导出等基本操作，以及实现一些 MATLAB、统计软件等功能。要满足以上要求，普通的商业监控软件和集中方式部署的简单监控系统已经力不从心，海量数据运维人员要有更强大的整体把控能力，使用 Nagios 的分布式部署特性和插件机制打造弹性扩展和不断进化的监控体系，对机房网络、带宽、硬件、服务器的性能进行实时监控，以及支持上层应用监控，实现数据分析等，做到对各个方面的情况了如指掌。

面对上万台机器，好几十个模块，几十万个监控项，想要了解哪些机器监控项缺少、哪些机器监控项异常、今天有哪些监控项报警、报警了多少次、团队中每个人每天收到多少报警、哪些是可以系统自动处理不报警的等，都需要从监控数据入手。Nagios 能够做到使运维

团队对整个平台的监控有直观而全面的了解，并在数据的指导下动态调整监控项和阈值，持续完善监控系统。

大规模的互联网公司都极其详细地定制化监控需求，具备自主开发监控系统的强烈渴望。而 Nagios 能够融入用户多年的运维监控经验，支持自主打造个性化监控系统，并且根据业务需求不断进行优化和完善，这正是商业监控系统做不到的。弹性的 Nagios 监控平台是一套统一的分布式监控平台，支持系统监控、网络监控、客户端监控、容量监控、趋势监控等，能自动添加基本监控，对服务器、虚拟机、应用 VIP、网络设备、Java 应用等能提供准实时预警、报警。在被动监控模式下，从数据采集到发出报警仅需要短短几秒钟，让运维人员第一时间掌握服务的健康状况。同时，它还具备多种故障预测及发现方式、丰富的数据图表展示、容量规划和报警，以及视图的定制等功能，是云环境和海量运维监控的最佳选择。

1.3.2 Nagios 的主机检测与服务检测

Nagios 的应用场景一般分为主机检测和服务检测。主机检测——即对一台服务器的检测，通常是使用简单的“ping”命令来检测主机是否存活。而服务检测就可以包罗万象了，从对网络服务，例如 HTTP、SMTP、DNS 服务等检测，以及进程检测、CPU 负载检测、日志文件检测等，通通都可以纳入服务检测的范畴。值得注意的是，Nagios 仅在认为有必要时才执行主机检测，例如，只有当某台被检测主机上的所有服务检测项都不可达时，Nagios 才执行主机检测，即使用 ping 命令检测该主机是否在线。否则哪怕在被检测主机上仅有一项服务可以被检测，而其他服务项均存在异常，Nagios 的灵活检测机制也会认为该主机是正常的，并不会使用 ping 命令执行对该主机的存活性检测并将主机判断为宕机状态。

最简单的网络服务检测手段包括查看指定端口是否开放，以及网络服务监听是否正常，但是并不能判断到被检测的网络服务是否真正运行正常，是否确实在对外提供服务。Nagios 却能在网络服务检测的百尺竿头上更进一步。例如，对 SMTP（简单邮件传输协议）服务而言，Nagios 能够检测到邮件服务器是否能够向客户端发送了“220”的输出信息，即 SMTP 问候信息（当邮件服务器上的 SMTP 服务接受一个客户端发起的 SMTP 连接的时候，它会向那台客户端发送一个问候信息，这些信息作为邮件服务器的标识，而且发送问候信息的目的就是告诉对方邮件服务器已经准备好了）。再比如，对于数据库服务而言，Nagios 能够检测到该数据库服务器是否能够解析并返回 SQL 查询请求。因此，在服务检测的深度和广度方面，Nagios 均能够做到锦上添花的效果。

Nagios 在设计之初，就考虑到了网络拓扑图中主机之间的依赖关系检测。在事先配置好主机依赖关系的前提下，如果目标主机宕机，Nagios 会将该主机标识为“不可达”状态，与之存在依赖关系的相关主机及相关服务都不会被检测，监控人员因而避免了无关告警信息的狂轰滥炸，可将注意力集中在关键主机及服务项的监控方面。另一方面，借助于依赖关系检测及告警，系统管理员也可以在纷繁芜杂的告警或者故障信息中准确定位根源故障，并有助于问题的第一时间解决。

1.3.3 监控信息的提供者

与众多传统监控工具相比，Nagios 的亮点在于其遵循完全开源的模式以及插件式的架构设计。Nagios 的核心逻辑并不包含任何的主机检测或者服务检测，相反，Nagios 使用名为 Plugin

(插件) 的一小段脚本或者小程序来执行检测。Nagios 提供了能够在各类操作系统平台上执行最基本类型检测的插件库, 可以执行操作系统 CPU、文件系统、内存、换页空间、网络服务等基本指标的检测。对于想掌握操作系统及相关进程基本运行信息的系统管理员来说, 这些检测项就已经足够了。如果还嫌不够, 想要关注更多主机上的服务状态, 或者想要关注关键业务的运行状态, 没问题, Nagios 提供了完美的答案——只要您有基本的编程知识, 并掌握一些编程语言的话, 完全可以编写自己的插件检测程序。但是在尝试定制化开发检测插件之前, 别忘了到网络上搜寻一下是否已经有类似的插件程序存在。经过这么多年发展, Nagios 已经具备大量拥趸, 开源世界的贡献者们已经为其开发了大量且多样的检测插件, 可以大大节省监控管理员的插件开发负担。在 <http://www.nagiosexchange.org> 上, 可以找到海量插件, 从主机到数据库、从网络设备到存储设备、从硬件检测到机房环境检测, 应有尽有。

Nagios 的检测插件通常是运行在被检测服务器上的一小段程序, 常见的是由脚本类编程语言, 例如 Bash、Perl、Python 等语言编写。Nagios 插件可以输出 OK、WARNING、CRITICAL、UNKNOWN 四类状态, 分别代表被检测项处于正常、警告、紧急以及未知四类运行状态。

插件式检测机制意味着 Nagios 可以检测任何 IT 系统——只要该系统能够支持脚本检测, 传递 OK、WARNING、CRITICAL 和 UNKNOWN 等状态信息给 Nagios。也就是说, Nagios 的检测对象在范围上没有任何限制, 只要系统管理人员能够找到一种途径让系统能够传递各类性能数据或者告警数据。例如, 可以将基于红外传感器的温度探测系统所发送的温度数据以及告警信息传递给 Nagios 监控系统, 从而实现机房温度探测和机房安全检测。

1.3.4 及时的通知机制

Nagios 设计了一个复杂而精巧的告警消息通知机制。借助于该机制, 管理人员可以设定, 当特定类型的通知消息 (主机或者服务的警告、紧急告警信息, 或者故障恢复信息等) 发生后, 并非系统里所有联系人员都会接收到该通知消息, 而是只有预先定义的联系人员组 (contact group) 里的人员才能够接收这些通知消息。同样地, 管理人员也可以在联系人组里设置多种级别的通知消息接收策略, 如进一步转发这些通知消息, 抑或是忽略这些通知消息。

在 Nagios 的消息通知机制中, 如果某项主机或者服务被全天候监控, 并不意味着系统管理员必须时刻准备接收通知消息而无法得到片刻休息。管理人员可以指定 Nagios 在每周的工作日——比如周一至周五的早 8:00 至下午 5:00——将告警信息通知到个人, 其余时间可以不通知。如果系统管理人员接收到告警通知消息, 在规定的时段内仍旧无法解决问题, Nagios 可以将该服务告警信息进一步通知到更高一级别的联系人, 这就是 Nagios 的告警消息通知升级 (Escalations) 功能。如果不采取通知升级功能, 问题就有可能一直暴露在工程师层面无法解决, 更高一级且拥有更多资源权限和管理权限的人员迟迟得不到通知, 导致后果的蔓延和恶化。

Nagios 能够运用自由定制的、多样化的外部通知机制, 将多种告警信息及时准确地传递给管理人员。借助于邮件系统、短信网关、语音服务器、微信等多种通知手段, 系统管理员可以第一时间从多种途径接收到 Nagios 系统传递的各类通知信息。

Nagios 不仅提供多样化的检测机制和通知机制, 还为系统管理员提供了 Web 界面, 有助于查看丰富的告警信息。无论管理人员想查看信息系统的监控状态总览, 或者技术人员想查

看单独的主机或者服务监控项的详细信息、主机组或者服务组的概要信息，Nagios 总是为不同角色的人员提供不同的监控视图，有助于一目了然地获知系统的各类故障。

系统管理员当得知某个主机监控项或者服务监控项出现问题，但短时间内不会造成影响的时候，可以直接联系在值班室的同事，将告警监控项设置为“已确认（Acknowledged）”的状态，标明该主机或者服务的告警信息已经被系统管理员确认为不会造成影响，以便于大家将注意力集中在其他关键监控项上。同样地，系统管理员还能够为主机或者服务监控项设置计划停机时间段，避免 Nagios 在此类时间段仍然发出不必要的告警信息。

Nagios 还提供历史日志查询功能。管理人员可以按照选定时间段浏览 Nagios 的各类运行日志信息——哪位系统管理员收到了告警通知消息，以及在过去的时间段内有哪些主机或服务产生过告警信息等等。

1.3.5 从外部系统接收信息

在前述的消息通知机制中，Nagios 使用手机短信、邮件服务等外部系统发送各类告警信息，但相反的路径同样可以。通过一个独立的接口，外部的独立系统也可以向 Nagios 发送状态检测信息，甚至是命令——从重启 Nagios 到各类检测指令无所不包。利用该反向机制，外部的独立系统可以向 Nagios 传递信息，如 Syslog 日志信息等。在外部系统集成方面，Nagios 没有任何限制。例如 Nagios 支持分布式监控，这也使得位于不同网络服务区域的多台 Nagios 服务器能够向中心 Nagios 服务器发送各自所辖网络区域内的主机服务和服务状态检测信息。统一的视图使得 IT 运维监控系统的分布式部署、统一监控不再是难以企及的梦想。

1.3.6 Nagios 与 Linux 的关系

Linux 操作系统是 UNIX 操作系统的一种克隆系统，它诞生于 1991 年的 10 月 5 日（这是第一次正式向外公布的时间）。Linux 借助于 Internet 网络，通过全世界各地计算机爱好者的共同努力，已成为今天世界上使用最多的一种 UNIX 类操作系统，并且使用人数还在迅猛增长。回到上世纪 90 年代，Mandrake Linux 还是唯一的 Linux 发行版，而今天，Linux 发行版数不胜数，这款操作系统现在有 100 多种。这也是开源软件具有的优点之一。

在 Linux 的诸多发行版当中，Ubuntu、Linux Mint 和 PCLinuxOS 被认为是新用户最容易上手的。而 Slackware Linux、Gentoo Linux 和 FreeBSD 是需要经过大量的学习后，才可以有效地加以利用的更先进的发行版。openSUSE、Fedora、Debian GNU / Linux 和 Mandriva Linux 操作系统一直遵循“中间道路”，意味着具备一定基础的用户可以选择这些发行版。而 Redhat Enterprise Linux 和 CentOS 是企业级的发行版，对于那些偏好稳定性、可靠性和高级尖端功能的企业级技术人员特别合适。

Nagios 最初设计为运行在 Linux 操作系统上，但因为 Linux 脱胎于 UNIX 操作系统，故其他来源于 UNIX 的操作系统，例如 Solaris、BSD、AIX 以及苹果公司的 Mac OS X 操作系统，都可以运行 Nagios。甚至有人声称在微软的 Windows 操作系统上安装了 Cygwin（一种能够在 Windows 平台上运行的 UNIX 模拟环境的工具）之后，也可以成功运行 Nagios。但笔者建议这种部署方式只能在测试环境下供爱好者研究所用，在正式的生产环境中，还是应该选择企业级的 Linux 操作系统。

在接下来的章节中，我们主要以 RedHat Enterprise Linux Server 6.5 X86_64 位发行版作为主要研究对象和 Nagios 的运行操作系统。

1.4 Centreon 简介

与 Nagios 一样，Centreon 也是一款著名的开源监控软件，由法国人在 2003 年开发。Centreon 的历史更为传奇，它最初是由默默无闻的公司新人（新人的创造力往往是惊人的，但很少像 Centreon 的开发者们这么能持之以恒地坚持一个项目并持续创新）在工作之余且没有预算的情况下开发的。但是现在，Centreon 已经走出法国，在全世界得到大范围的应用，甚至出现在著名 IT 咨询公司 Gartner 的报告中（<http://blogs.gartner.com/jonah-kowall/2012/10/03/low-cost-and-free-monitoring/>）。

或许是由于 Nagios 项目的成功，在项目创建之初，无论是在创始人心目中，还是在众人的眼中，Centreon 还是作为“Nagios 的 Web 前端管理界面”而出现的，是 Nagios 的众多用户界面插件中毫不起眼的一个小角色。但随着 Centreon 项目的演化，其变得日渐强大，逐渐成为一套完善而又齐全的 IT 运维监控解决方案，同时具备了优异而又稳定的用户界面和活跃的开源用户社区，成为开源监控系统中的一位重量级玩家。

与 Nagios 简陋的 Web 用户界面有所不同，Centreon 拥有完整的企业级 IT 运维监控管理界面。Centreon 提供了监控系统的实时监控界面和后台管理界面，并辅以各种类型的图形、表格、图标，用来显示监控平台搜集到的各类性能数据，以及平台本身的运行统计数据。Centreon 提供了各种类型的模板，帮助用户批量创建各类监控模型，避免了在 Nagios 上需要逐一创建的大量重复劳动。与一切依赖手工配置的 Nagios 相比，Centreon 提供了建设企业级监控系统所需要的一切自动化工具和设施，这正是 Centreon 得到更大范围运用的根本原因。

Centreon 在 IT 运维监控方面正在创建一种模式——逐渐替代曾经被认为是不可替代的商业监控系统，实施成本低廉且固定，不会因部署范围的扩大同步增加费用。随着社区版用户的增多，Centreon 正在形成一种自下而上的互联网文化，在这个开放社区里，活跃用户的意见得到了充分倾听，有价值的意见在新的 Centreon 发行版本里得到采纳。例如，Centreon 的配置界面集成了全新的模式，引入了用户模板、各类主机和服务监控模板、自定义宏、模板之间的关联和继承等等，这一系列的改进有助于用户创建更多的监控服务和更大规模的部署。此外，Centreon 的用户界面得到了进一步改善，提供了基于关键字的监控项过滤、基于组的过滤、停机维护时间管理和告警日志管理等，Centreon 甚至还提供了报表功能，支持报表导出为 CSV 格式。

遵循开放模式带来的第一项好处在于，Centreon 能够集成多种必要组件，包括多个 Nagios 服务器，从而构建一个分布式的企业级 IT 运维监控系统。Centreon 可以连接多个 Nagios 服务器，将来自多个源头的告警信息汇聚到统一的监控告警视图中。Centreon 的监控项配置功能可以在多个 Nagios 服务器之间保持配置文件的同步，避免了 Nagios 分布式架构中配置文件不统一的问题。与 Nagios 的多样化插件机制促使自身保持灵活开放的模式一样，Centreon 的分布式管理机制能够使其管理下的多个 Nagios 服务器之间保持监控任务的负载均衡，从而使得 Centreon 在管理多样化监控项及分布在不同地理位置的服务器监控方面表现优异。

该模式的第二项明显优势在于，Centreon 能够基于自身的访问控制列表（Access Control

List, ACL) 实现精细化的访问管理, 通过设置安全策略来保障用户只能看到被授权访问的资源, 从而实现自定义的访问控制。相较于 Nagios 采用简单的、基于 Apache 的网页用户授权和认证机制相比, Centreon 的访问控制机制更能满足企业级 IT 系统的信息访问控制策略, 在满足大型企业信息安全及业务访问需求方面无疑更胜一筹。

Centreon 的灵活性及用户成熟度使其成为 IT 运维监控人员眼中的佼佼者, 其部署规模稳步增长。围绕着 Centreon, 用户群逐渐分化为两种: 一种是追求完全自定义界面的用户, 另一种则是追求更广泛监控信息和监控报表的用户, 前者追求华丽的外表、后者则更注重翔实的内容, 可谓相得益彰。

在 Centreon 开源社区版的基础上, 其开发团队还陆续推出了 Centreon MAP、Centreon BAM 和 Centreon BI 等商业产品。Centreon MAP 是 Centreon 的可视化监控视图界面, 允许系统管理员以图形的方式定制自己的可视化监控界面。Centreon MAP 可以定义类似于微软 Office Visio 类似的网络拓扑图、应用拓扑图、地图以及业务流程等逻辑或者物理视图, 可以直观地为用户呈现业务流程全景, 有助于用户在故障时准确定位故障点。而借助于 Centreon BAM, 用户可以定义业务流程中的关键节点, 实现关键业务流程的风险监控, 进而完善企业信息服务, 例如 ERP 服务的高可用性。最终, 由 Centreon BI 向用户提供性能报表和设备故障率统计, 用于后续的容量管理和问题管理。在 Centreon 的生态环境中, 从底层检测数据的采集, 到中间数据逻辑处理, 再到前台展示, 各个层级都能够通过灵活的界面进行配置, 便于用户能够从业务视图、逻辑拓扑、设备管理、告警统计、业务流程展示等角度对 IT 运维监控工作做全方位展现, 实现了“一屏在前, 全局尽显”。

限于 Nagios 的开发团队在推陈出新上的乏善可陈, Centreon 团队在两个方面做到了改进, 站在了巨人的肩膀上。改进之一是基于原有的 Nagios 核心引擎打造了新一代的 Centreon-Engine 引擎, 改进之二是创建了 Centreon 代理程序, 用以替代 Nagios 的 NDOUtils 工具。

1.4.1 Centreon 引擎

Centreon-Engine, 又称 Centreon 引擎, 是在 Nagios 核心引擎基础上演进而来的另外一个分支。其目的在于, 通过发布一系列 Nagios 核心引擎的补丁, 丰富 Nagios 核心引擎的特性。其相较于 Nagios 引擎的核心变化在于改进 Nagios 性能、解决 Nagios 长久以来被忽视的 bug, 以及添加各类丰富的特性以满足用户期待等等。

1.4.2 为什么要有 Centreon 引擎

在过去的数年中, Centreon 从 Nagios 中获益良多, 围绕着这两款软件周围, 产生了诸多活跃的用户群体, 以及他们所做的一系列杰出工作。但是, 在过去的 3 年中, 这股激情和活力在 Nagios 方面明显下降。Nagios 的开发者们选择了另外一条道路, 倾向于借助 Nagios 向用户提供专业的监控服务, 换句话说, Nagios 不那么自由了——尽管仍然遵循开源模式, 但是变得越来越商业化。当然, 换个角度看, 这种变化未尝不是好事, 一方面它证明了 Nagios 在当今世界的用户群体越来越广泛, 另一方面专业的服务也利于 Nagios 的进一步推广。但是, 对于开源世界和自由社区来讲, Nagios 越来越晦暗不明的开源定位和前景使它的号召力未免越来越暗淡无光。

至今,自 Nagios 的第 3 版于 2008 年推出已经过去了近 6 年时间,在这 6 年时间里,Nagios 的改进十分有限。对于精通 C 语言且关注 Nagios 的开发人员来说,由于 Nagios 核心开发团队的封闭性和排他性,很少能够参与到 Nagios 的改进工作中,这不能不说是一个遗憾。

开源社区的卓越用户们很久以来就持续关注该问题,包括 Centreon 的开发团队,以及其他 Nagios 的分支版本,如 Icinga 等,都在尝试着推动 Nagios 的改进。用户愈加认识到,Nagios 的迟迟不更新已经影响到了它的进一步推广,一些重要的特性必须体现在 Nagios 中,才能满足用户的要求。但遗憾的是,由于 Nagios 开发团队的封闭性,这些具备激情和技术能力的开发者们的意见并不能得到立即采纳。即使是到了 2013 年 9 月 20 日,Nagios 推出了其正式的 4.0 版本,但由于存在一些不足甚至不合时宜的特性,也不能完全令用户满意。

由于 Nagios 面临的挑战持续存在,而开发者们的意见又迟迟得不到采纳,故开发者们将热情转移到了 Nagios 的分支版本中,产生了诸如 Centreon-Engine 等 Nagios 的派生版本。无论是直接修改 Nagios 官方的源代码,抑或是重新派生出新的 Nagios 分支版本,至少在这些分支版本中开发者们的意见得到了充分的体现。

1.5 NagVis 简介

截至 2013 年,Nagios 已经走过了 11 个年头,在这 11 年中,Nagios 已经赢得了巨大的成功、收获了广泛的赞誉,但在这个社交媒体和移动互联网大行其道的关键时刻,一些人也敏锐地意识到,Nagios 的用户界面十几年来几乎没有任何变化,已经无法满足普通用户挑剔的眼光了,急需升级更新。如果 Nagios 社区不能提供对用户友好的监控界面,Nagios 的进一步推广将会非常困难。面对这些批评,Nagios 的开发者们仍旧坚持朴素的、非商业化的用户界面,甚至将其作为 Nagios 的传统而在后续的作品中传承。

但对于用户意见一向反应敏感的开源社区却坐不住了,在推出了 Nagios 的管理工具——Centreon 之后,开源社区再一次推出可以与商业监控系统视图管理工具相媲美的 Nagios 平台界面展示插件(addon)——NagVis。该插件可在一张监控地图上自由组合并展示 Nagios 收集到的性能信息和故障告警信息,为习惯了 Nagios 传统的 CGI 界面的人们带来了眼球上的震撼。

NagVis (<http://www.nagvis.org/>) 是 Nagios 社区群体中的一个重要插件(在 Nagios 的术语中称为 addon,不是 Plugin),负责将 Nagios 的监控信息可视化,即向用户展示 Nagios 的监控信息。在企业级 IT 监控中心的大屏幕上,我们往往可以看到 NagVis 的身影。NagVis 的独特之处在于,它可以允许用户自由选取并上传一张自定义的背景图片(在 NagVis 术语中,称为“地图”,即 maps),并且能够在背景图片上摆放各类图标——代表被监控主机或者被监控服务,每个图标都可以展示各自代表的被监控主机或者服务的实时状态。如此一来,用户通过观察图标的颜色,就能及时了解到系统的最新状态。

如此一来,NagVis 借助简单而又实用的设计迅速虏获了大批的拥趸,为用户带来了各式各样的监控界面设计方法。例如,用户可以将一张地图作为背景,在地图上相应的地点建立机房、机柜、服务器图标,在服务器故障之后就可以做到迅速定位。又比如,用户可以借助 Visio 等建模软件设计业务逻辑图作为背景,再引入相关联的服务器图标和服务监控图标,共同构成对于关键流程的监控视图。当服务器或者服务发生告警后,用户可以第一时间感知故

障，并借助流程图判断业务受影响的范围和程度，如图 1-1 所示。

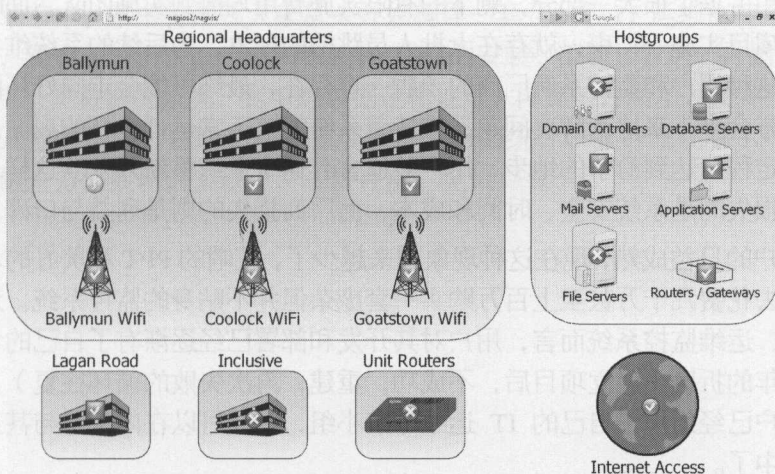


图 1-1 NagVis 监控视图样例

1.6 为什么要基于开源软件构建 IT 运维监控系统？

要回答此问题，我们首先要思考一下商业 IT 运维监控解决方案存在的种种缺陷。

与基于开源软件所构建的监控系统之灵活多样性相比，商业监控软件企图提供的是一套覆盖范围广、监控项目全的一系列解决方案。商业监控软件首先假设所有人需要的是同一套解决方案。就某种程度而言，确实存在诸如此类的需求——尽管企业部署了多种多样的 IT 系统，但都希望在某些服务宕机后及时得到通知。因此，商业监控软件厂商为了卖出更多的监控软件，自然希望自己的产品能够包罗万象，监控范围囊括世上所有已知的软件或者硬件设备。能够监控的设备越多，潜在的客户就越广，最好将让自己的监控产品做成包罗万象的服务，使自己与客户的交易达到一锤子买卖的效果，这是大多数厂商希望达到的目的。

在工作实践中，大家可能会遇到如下的现象，用户考虑到自己的业务越来越依靠 IT 系统的 24 小时不间断运行，迫切需要上马一套监控软件来帮助自己监控信息系统，甚至能够预测系统未来的运行趋势；同时用户了解到某几家厂商的商业监控软件在业内很有名气，于是分别向对方发送了邀请函，希望通过对比的方式选择适合自己的 IT 运维监控产品，并希望立竿见影地见到系统实施的成效。只要软件厂商符合条件且对用户的需求感兴趣，大多会派几名市场人员和售前工程师抵达用户现场，打开花哨 PPT 的同时向用户不遗余力地推销自家的产品，做出一个大而具有诱惑力的实施方案，对于用户的需求一概说 YES。用户自然喜出望外，与心仪的厂商一拍即合，共同签订了合同——当然分为两部分，软件授权费用（一般按照监控节点数购买）以及实施费用。但是到了实施阶段，项目往往陷入了僵局，说好的种种承诺没有兑现，规划好的实施范围开始缩水。甚至隔了几年之后，用户都找不到当初的实施人员协助自己解决问题——用户惊讶地发现该公司的项目团队已经解散！

这是一种矛盾，用户和监控系统厂商两者之间都存在的问题。从用户角度来讲，由于自身存在专业技术上的不足，以及对需求理解的不全面、不深刻，往往对即将实施的系统存在幻想，希望一劳永逸地解决问题，偏偏没有意识到监控系统的成熟度往往是伴随着使用

者对自身信息系统了解程度的成长而不断增长的。从厂商的角度而言，用户的预算已经被监控软件授权费用占据了很大一部分，剩下的有限实施费用均摊到实施团队，面临高额的人员成本。往往是项目实施一结束，就存在大批人员跳槽的现象，为后续的系统维护升级工作带来诸多不便。这种用户和监控系统厂商的矛盾一直存在，最终可能会走向好坏两种局面——好的局面是双方会坐下来协商解决问题，维持着系统能够正常运行；坏的局面是双方的矛盾达到积累到一定程度达到纷争的地步，用户被迫舍弃这个系统重新开发。这样一来对双方都有损失，用户损失的是系统安全、时间和成本，而厂商损失的则是利益与口碑。

伴随着用户的日益成熟，现在这种现象越来越少了，花哨的 PPT 和美好的承诺再也不能轻易忽悠用户去花费几十万甚至上百万购买一套庞杂但并不贴身的监控系统。对于需要时时刻刻关注的 IT 运维监控系统而言，用户对其开发和部署已经逐渐有了自己的考虑。也许就是因为这许多年的折磨（建立项目后，不成功，重建，再次失败的循环往复），以及从中汲取的教训，用户已经组建了自己的 IT 运维专家小组，已经可以在内部参与甚至主导这种系统的开发工作中了。

伴随着这种演变，用户的技术团队对于监控系统的理解和要求也越来越高。用户逐渐了解到监控系统并非想象中的一次性交钥匙工程，而是在实施之前需要大量的客户化和自定义需求的工程。用户逐渐意识到，商业监控系统厂商能够提供的，也许并非想象中的可以随心所欲监控任何设备和软件的完美监控系统。与其花费精力选择商业监控软件，考察它们哪个能够监控更多的系统，不如将注意力转移到如何打造一个适合自身的个性化监控系统，以及如何更有效地监控自己所关注的项目。

以监控系统中常用的检测网络是否通畅的 Ping 命令为例，商业监控系统都采用向目的地发送一个 ICMP（Internet Control Messages Protocol，即因特网信报控制协议）报文，并向用户报告是否收到一个应答消息，来检测目标服务器是否在线。但假如用户想进行多样化的 Ping 检测，例如向目的地发送指定数量的 ICMP 报文，或者想根据 Ping 命令结果的 RTT 时间长短报警而非根据主机在线与否报警。更复杂的检测命令还包括使用 IPv6 Ping 命令进行检测，或者在使用 Ping 命令检测之前先进行端口试探工作。以上个性化甚至定制化的 Ping 检测都超出了商业监控软件的能力，以至于受到了绝大多数商业监控软件的忽略。

从监控系统而言，用户愈加认识到对监控项进行个性化定制的重要性。而 Nagios 作为杰出的开源监控软件，其关注点正在于无与伦比的灵活性。Nagios 使用简单的、被称为“插件（Plugin）”的小程序或一小段脚本对用户的业务逻辑、软件、硬件、甚至机房环境进行全方位，自定义的检测。相比商业监控软件之间进行的“军备竞赛”而言，Nagios 的监控武器更为犀利，往往一招致命。一旦 Nagios 用户有了监控新设备和新软件的需求，新的监控插件就会迅速出现，提供比商业监控软件更快更精准的监控功能。实际上，Nagios 能够监控你能想到的任何事物，用户需要做的仅仅是配置管理而已。

选择 Nagios 作为企业级监控平台，意味着你的监控工作只能被想象力、技术能力和眼光见识所拘囿，换句话说，只要你选择 Nagios，只有你想不到的，没有后者做不到的。但同时也别忘了，相比商业监控软件轻而易举的安装配置过程而言，伴随 Nagios 的却是一条充满荆棘之路。随着 Nagios 的安装，你会发现没有任何可供选择的选项，事实上，Nagios 自身并不了解“如何”监控，而是希望用户自己能够对监控需求和监控手段有深刻了解，从而告诉 Nagios 去监控。换句话说，Nagios 的灵活性体现在用户需要自己选择检测插件或监控插件，

必要时编写适合自己的插件，来部署到 Nagios 上实现自己想要的监控。

接下来的章节中，本书将带领大家从头开始构建一套企业级 IT 运维监控系统。从 Linux 操作系统安装，到 Nagios、Centreon 和 NagVis 软件的安装、部署、管理，再到监控系统构建方法论，从底层检测数据的采集，到中间数据逻辑处理，再到前台展示，最终完成系统构建，便于用户能够从业务视图、逻辑拓扑、重要设备、告警统计、业务流程展示等多角度对 IT 运维监控工作做全方位展现，实现了“一屏在前，全局尽显”。

企业级 IT 运维监控系统的构建—— 从源代码到企业级系统

在 Linux 操作系统中，有一种系统软件，它的功能类似于 Windows 里面的“添加/删除程序”，但是功能又比“添加/删除程序”强很多，它就是 Red Hat Package Manager(简称 RPM，软件包管理工具)。此工具包最初是由 Red Hat 公司推出，后来被其他 Linux 开发商所使用，被广泛应用于在 Linux 上安装、删除软件。

根据 Nagios 的官方网站(<http://www.nagios.org/docs/pdrtoc/chapter1.html>)，Nagios 已经包含在一些流行的 Linux 发行版中以及它们的安全扩展功能包中以软件包(package)的形式存在，这些有 Ubuntu、Fedora、SUSE 以及 Debian 等著名的 Linux 操作系统。如图 1-1 所示，列出了目前包含 Nagios 包的所有 Linux 操作系统。

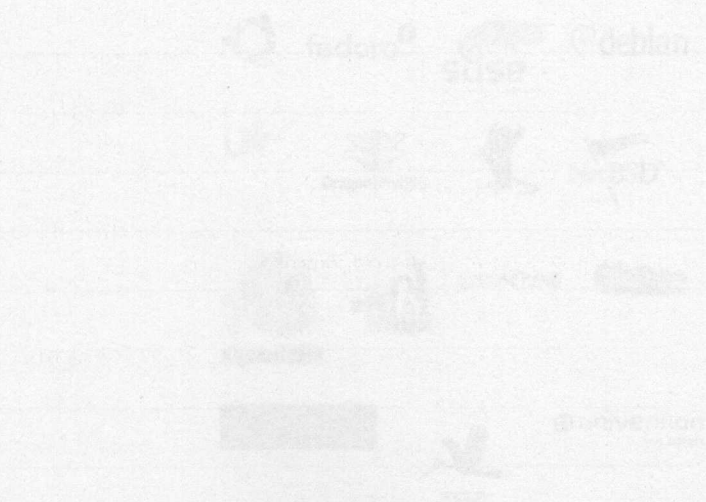


图 1-1 目前包含 Nagios 软件包的 Linux 操作系统列表



对于自身信息系统的了解程度而调整监控策略。Nagios 4 能够根据系统配置和性能指标，自动调整监控策略，以便更好地适应不同的系统需求。例如，对于高负载的系统，Nagios 4 可以增加监控频率，以便及时发现和处理问题。对于低负载的系统，Nagios 4 可以减少监控频率，以降低系统负担。

伴随着用户对网络监控的需求日益增长，Nagios 4 在市场上的地位也越来越重要。它不仅能够满足用户对网络监控的基本需求，还能够提供定制化的监控方案，以满足不同用户的需求。因此，Nagios 4 已经成为网络监控领域的热门选择。

对于企业来说，Nagios 4 的价值不仅仅在于它能够监控网络设备的运行状态，更在于它能够帮助企业及时发现和处理网络故障，从而保障业务的正常运行。通过 Nagios 4，企业可以实现对网络设备的实时监控，及时发现异常情况，并采取相应的措施进行修复。这不仅能够提高网络的稳定性和可靠性，还能够降低企业的运维成本。

此外，Nagios 4 还支持多种插件和扩展，可以根据用户的需求进行定制。例如，企业可以安装 Nagios 4 的插件，以实现对特定设备的监控。或者，企业可以利用 Nagios 4 的扩展功能，将监控数据与其他系统进行集成，以实现更全面的网络管理。

总的来说，Nagios 4 是一款功能强大、灵活多变的网络监控工具。它不仅能够满足企业对网络监控的基本需求，还能够提供定制化的监控方案，以满足不同用户的需求。因此，Nagios 4 已经成为网络监控领域的热门选择，也是企业保障网络稳定性和可靠性的有力工具。

第 2 章

企业级 IT 运维监控系统的构建—— 从源代码到企业级系统

在 Linux 操作系统中，有一种系统软件，它的功能类似于 Windows 里面的“添加/删除程序”，但是功能又比“添加/删除程序”强很多，它就是 Red Hat Package Manager(简称 Red Hat 软件包管理工具)。此工具包最先由 Red Hat 公司推出，后来被其他 Linux 开发商所借用，被广泛应用于在 Linux 下安装、删除软件。

根据 Nagios 的官方网站(<http://www.nagios.org/about/propaganda/distros>)，Nagios 已经在一些通用的 Linux 发行版中以及它们的安全扩展功能包中以软件包(package)的形式存在，这包括 Ubuntu、fedora、SuSe 以及 debian 等著名的 Linux 操作系统。如图 2-1 所示，列出了目前包含 Nagios 软件包的所有 Linux 操作系统。

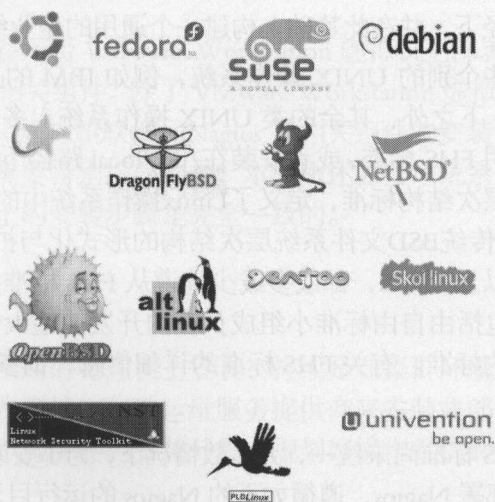


图 2-1 已经包含 Nagios 软件包的 Linux 操作系统列表

安装 Nagios 的最简单方式就是借助于以上 Linux 发行版中的软件包安装工具，这些工具已经帮您完成所有准备工作，例如 Nagios 所需依赖软件包的安装、Nagios 安装路径的设置等等。然而正因如此，这些安装工具往往已经设置了各自的 Nagios 安装路径，这些路径可能与 Nagios 源代码中指定的默认安装路径截然不同，这会导致后续一系列组件安装过程中都需要设置特别的路径，造成不便。

基于以上情况，基于最新的 Nagios 3.0 版（在本书编写过程中，最新的 Nagios4.0 版本已经问世，为稳定起见，本书采用的 Nagios 版本为 3.4.3）构建企业级 IT 运维监控系统的最好方式是从头开始，逐步安装每一个组件。从选择合适的 Linux 发行版开始，到 MySQL 数据库安装、Nagios 安装（包括 NDODB、NRPE 和 NSClient++等组件的部署）、Centreon 安装、NagVis 的安装，以及被监控主机和监控项的配置等，直至最后形成一套完整的企业级 IT 运维监控系统。从无到有构建系统的这一过程中，充满了艰辛，也蕴含着解决问题的快乐。

当用户编译自己的 Nagios 软件，从零开始构建 IT 运维监控平台的时候，用户就成了独一无二的平台架构师——对软件架构了如指掌，每一项目录结构、每一项参数配置，无不妙手搭配。用户可以自由掌握监控平台中每一项组件的配置，如果对于某项组件的性能不满意，可以自由升级或者回退已安装的组件，而不必依赖 Linux 操作系统的缓慢升级。通过从源代码开始安装并构建监控平台，可以体验到开源软件的魅力，以及为人们带来的无穷探究乐趣。

2.1 可供选择的操作系统

Nagios 从一开始就构建在 Linux 操作系统上，但是由于 Linux 和 UNIX 系统天然的关联特性，Nagios 同样可以运行在 HP-UX、Solaris、IBM AIX、甚至 MAC OS X 等诸多类 UNIX 操作系统上。在上述已知的操作系统中安装 Nagios 的主要区别就是安装路径问题，不同的系统根据各自的目录结构，会将 Nagios 自动安装在不同的目录下，要保持安装路径的一致非常困难。即便是不同发行版本的 Linux 操作系统，由于彼此之间的文件系统也存在很大区别，Nagios 的运行目录也会被自动放置在不同的目录下。因此如果想保持目录的一致性，避免无谓的目录改变，最好的方式就是从源代码开始，编译并安装 Nagios 软件，使其位于默认的 /usr/local 路径下，并在此基础上构建一个通用的企业级 IT 运维监控系统。

除了一些个别的 UNIX 操作系统，例如 IBM 的 AIX 系统，会将开源软件安装在路径 /opt/freeware 下之外，其余的类 UNIX 操作系统大多会选择将 Nagios 以下面两种路径方式安装——使用 FHS 标准，或者安装在 /usr/local 路径下。FHS (Filesystem Hierarchy Standard)，即文件系统层次结构标准，定义了 Linux 操作系统中的主要目录及目录内容。在大多数情况下，它是一个传统 BSD 文件系统层次结构的形式化与扩充，多数 Linux 发行版，例如 Red Hat、Mandriva，以及 SuSE，都或多或少地遵从 FHS 标准并且声明其自身政策以维护 FHS 的要求。然而，包括由自由标准小组成员在内开发的绝大多数发行版（截至 2009 年），并不完全执行建议的标准。有关 FHS 标准的详细信息，请参考维基百科上的“文件系统层次结构标准”定义。

由于 FHS 标准尚未统一，大多数情况下，为了安装和维护的便利，我们会选择从源代码开始编译和部署 Nagios，遵循如下的 Nagios 的运行目录结构，如表 2-1 所示。

表 2-1 自源代码编译并安装后的 Nagios 目录结构

文件类型	文件路径
配置文件（Configuration files）	/usr/local/nagios/etc
页面文件（HTML）	/usr/local/nagios/share
网关接口（CGI）	/usr/local/nagios/share
Nagios 主程序及其他可执行文件 （Program daemon and other executables）	/usr/local/nagios/bin
锁文件及队列文件（LockFiles 、FIFO）	/usr/local/nagios/var
日志（Logs）	/usr/local/nagios/var
插件程序（Plugins）	/usr/local/nagios/libexec

2.1.1 选用 Red Hat Enterprise Linux 作为操作系统

在本书中,我们选择 Red Hat Enterprise Linux 作为整个企业级 IT 运维监控平台的操作系统。Red Hat 应该说是在国内使用人群最多的 Linux 版本,甚至有人将 Red Hat 等同于 Linux,而有些资深系统管理员更是只用该发行版本的 Linux。Red Hat 系列的包管理方式采用的是基于 RPM 包的 Yum 包管理方式,包分发方式是编译好的二进制文件。稳定性方面非常出色,适合核心服务器使用。

Red Hat Enterprise Linux 6 发行版内置有 GCC 4.4 编译器、OpenJDK 6、Tomcat 6、Ruby 1.8.7 和 Rails 3、PHP 5.3.2 与 Perl 5.10.1,数据库前端有 PostgreSQL 8.4.4、MySQL 5.1.47 和 SQLite 3.6.20,这些常用组件和操作系统紧密结合在一起,构成了一个极佳的 Nagios 服务器端开发、部署及运行环境。

2.1.2 选择部署方式

为了方便和便于演示,本书选择在一台 VMware Workstation 虚拟机上部署企业级 IT 运维监控系统,安装 Nagios 软件及其他相关组件。除了 VMware Workstation 虚拟机软件外,您还可以选用另外一款著名的 VirtualBox 虚拟机软件。Nagios 及相关组件的安装步骤在这两款虚拟机软件之间仅存在细微的差别。需要特别指出的是,本书所述监控系统系列软件在虚拟机上的步骤和在一台实际物理服务器上的安装步骤是完全一致的。

2.2 服务器安装规划

无论是在虚拟机上安装,还是在实际的物理机上安装,抑或是在集群环境中部署,必要的话,还是应该请具备专业经验的操作系统安装和运维服务提供商来安装或部署。如果不具备这样的条件,那么事先详细规划一下服务器的安装计划,规划好操作系统的参数设置等等是十分必要的。

2.2.1 服务器参数规划

以下是在安装一台新的 Nagios 服务器时需要事先规划的参数：

机器名

internet 域名（Internet Domain）。即与该服务器相关联的唯一、通用的网络域名，可供在 internet 上区别该服务器。如果服务器放置在企业内部网，则该项可忽略。

- 服务器 IP 地址、服务器子网掩码、服务器网关。
- 内网 DNS 服务器 IP 地址、外网 DNS 服务器 IP 地址。
- 操作系统管理员用户名和密码。
- Nagios 管理员用户名和密码。
- Nagios 的 Windows 客户端代理——NSClient++端口号（默认 12489）。
- Nagios 的 Unix 操作系统客户端代理——NRPE 端口号（默认 5666）。

除了以上参数之外，还应该了解操作系统的防火墙配置、路由配置、集群配置（如果安装有 Linux 集群软件或其他集群软件）等等与服务器相关的重要配置信息，必要时请查看操作系统管理员手册或者联系操作系统管理人员。

2.2.2 服务器存储规划

以上即是安装一台服务器需要提前考虑的参数设置，这些参数中的大多数都可以在安装之后修改，如果不合适，还有补救的机会。事实上，一旦决定开始安装操作系统，最需要重点考虑的是存储的分区问题，即如何划分存储空间、创建文件系统、规划文件安装目录等。

试想一下，如果是服务器存储空间不足或者文件系统划分不当，后续在监控服务器投产后，不得不采用删除文件的方式来释放磁盘空间，这是任何一位负责的系统管理员所不能容忍的。为了避免这种窘境，除了事先规划好每个文件系统上需要部署哪些应用程序、估算应用程序各自对安装空间和剩余空间的要求，以及应用程序对文件系统的读写特性要求之外，还要对服务器投入运行之后，对于每个文件系统的动态增长，以及程序读写特性等方面持续不断地做精细的调优工作。

一般来说，规划 Linux 服务器文件系统及划分服务器存储空间时需要遵循下列规则：

- 避免在根分区（root volume）安装占用空间较大或者增长过快的应用程序。
- 相应地，尽量将每类应用程序安装在除了根分区之外的其他不同的分区。

和 Windows 操作系统里 C 盘剩余空间不能过小一样，对于 Unix 操作系统或者 Linux 操作系统而言，根分区空间划分过小，以及在根分区存放容易引发占用空间增长的应用程序都是非常危险的举动，容易导致操作系统故障，甚至无法正常启动。对于实时提供访问的服务器而言，这是不可接受的。另一方面，将用途不一样的应用程序分布在不同的文件系统里，有利于故障的隔离和空间的灵活分配，例如，为占用空间增长较快的应用程序分配空间更大、数量更多的逻辑卷等等。必须谨记，服务器的各项资源是宝贵的，且投产之后再调整服务器配置的代价十分高昂（需要很多的风险考量及冗长的管理审批工作），必须事先做好充分的规划，这一点十分重要。

2.3 Linux 的逻辑卷 (LVM) 管理机制

LVM 是逻辑卷管理 (Logical Volume Manager) 的简称, 它是 Linux 环境下对磁盘分区进行管理的一种机制, LVM 是建立在硬盘和分区之上的一个逻辑层, 来提高磁盘分区管理的灵活性。通过 LVM 系统管理员可以轻松管理磁盘分区, 如: 将若干个磁盘分区连接为一个整块的卷组 (volume group), 形成一个存储池。管理员可以在卷组上随意创建逻辑卷 (logical volumes), 并进一步在逻辑卷上创建文件系统。通过 LVM, 管理员可以方便地调整存储卷组的大小, 并且可以对磁盘存储按照组的方式进行命名、管理和分配, 例如按照使用用途进行定义: 文件系统目录 `/usr/local` 和 `/var/log`, 而不是使用物理磁盘名 `sda` 和 `sdb`。而且当系统添加了新的磁盘, 通过 LVM, 管理员就不必将磁盘的文件移动到新的磁盘上以利用新的存储空间, 而是跨越磁盘直接扩展卷组和文件系统即可。

2.3.1 为什么要使用 LVM

在为系统分区时, 如何精确评估和分配各个硬盘分区的容量是一项重要工作。因为系统管理员不仅要考虑到当前某个分区需要的容量, 还要预见该分区以后可能需要的最大容量。如果估计不准确, 当遇到某个分区容量不够用时, 管理员可能甚至要备份整个系统、清除硬盘、重新对硬盘分区, 然后恢复数据到新分区。

虽然现在有很多动态调整分区的工具可以使用, 例如 `Partation Magic` 等等, 但是它并不能完全解决问题, 因为分区容量可能会再次被耗尽。此外使用分区工具调整完毕磁盘容量后, 需要重新引导系统才能生效, 而对于很多关键的服务器, 停机是不可接受的。而且此类分区调整工具在管理跨越多个硬盘驱动器的文件系统时存在一系列问题, 影响系统安全。

因此完美的解决方法应该是在不停机的前提下, 自如地对文件系统的大小进行调整, 能够实现跨越不同磁盘和分区管理文件系统。幸运的是, Linux 提供的逻辑盘卷管理 (LVM, Logical Volume Manager) 机制就是一个完美的解决方案。简要说来, LVM 可以在不停机的情况下动态调整各个分区的大小, 并且保持原有的文件系统不变。

2.3.2 LVM 基本概念

前面谈到, LVM 是在磁盘分区和文件系统之间添加的一个逻辑层, 为操作系统屏蔽了底层物理磁盘布局, 并为操作系统及其用户提供了抽象的逻辑磁盘, 便于用户在这些逻辑磁盘上建立文件系统。

首先我们介绍以下几个 LVM 术语:

- 物理存储介质 (The physical media): 这里指系统的存储设备——硬盘, 如: `/dev/hda`、`/dev/sda` 等等, 是存储系统最底层的存储单元。
- 物理卷 (physical volume): 指的是物理硬盘, 或者与物理硬盘具有同样功能的设备 (如 RAID、存储设备等)。由于物理卷被 LVM 所管理, 因此包含有与 LVM 相关的管理参数, 是基本的物理存储介质。
- 卷组 (Volume Group): 由一个或多个物理卷组合而成, 可以在卷组上创建一个或多个逻辑卷。

- 逻辑卷 (logical volume) : LVM 的逻辑卷类似于非 LVM 系统中的硬盘分区, 在逻辑卷之上可以建立文件系统(比如/home 或者/usr 等)。
- PE (physical extent) : 每一个物理卷被划分为称为 PE 的基本单元, 具有唯一编号的 PE 是可以被 LVM 寻址的最小单元。PE 的大小是可配置的, 默认为 4MB。
- LE (logical extent) : 逻辑卷也被划分为 LE, 同样是可被寻址的基本单位。在同一个卷组中, LE 的大小和 PE 是相同的, 并且一一对应。

可以看到, 物理卷 (PV) 被由大小等同的基本单元 PE 组成。一个卷组由一个或多个物理卷组成, PE 和 LE 有着一一对应的关系。逻辑卷建立在卷组上, 逻辑卷就相当于非 LVM 系统的磁盘分区, 可以在其上创建文件系统。

到这里我们可以看出, 原本是直接在硬盘上创建分区, 然后在分区上创建文件系统。使用了 LVM 后, 在其中插入一个逻辑层, 相当于是在一块逻辑硬盘上创建逻辑分区, 然后在逻辑分区上创建文件系统。

新插入一个逻辑层, 对单个硬盘的读写会有一定的性能损失, 但其带来的好处是巨大的。首先, 逻辑分区大小不再受硬盘实际大小的限制, 它可以扩展到几块硬盘上; 其次, 逻辑分区可以很方便的做调整大小、备份等维护操作; 而且, 如果系统中存在多块硬盘, 通过设置逻辑卷到物理卷的映射关系 (采用 LVM striped mapping), 可以提高 I/O 的读写性能, 因为此时的读写是在多块硬盘上并发进行的, 比对单个硬盘的读写显然要快很多。

如图 2-2 是一个 LVM 使用的例子:

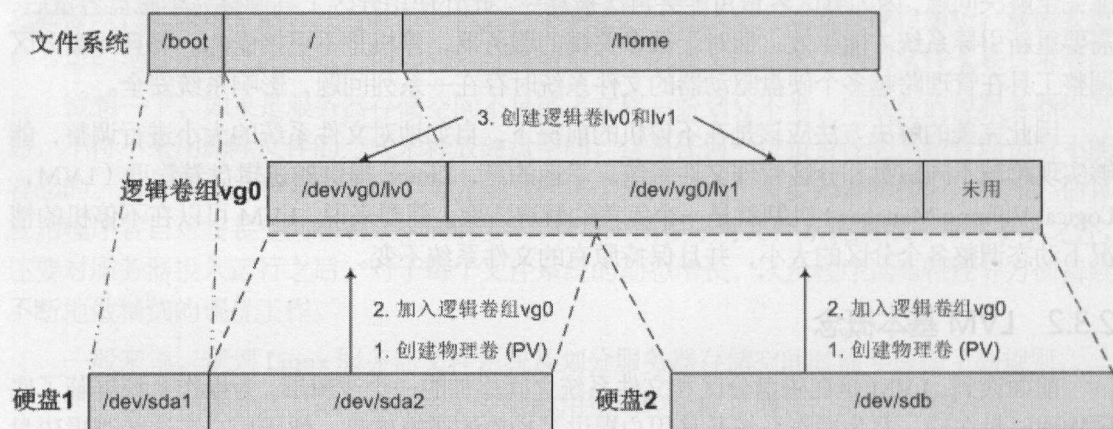


图 2-2 Linux 逻辑卷管理 (LVM) 例子

在图 2-2 中有两块硬盘, 其中硬盘 1 分了两个分区, `/dev/sda1` 和 `/dev/sda2`, 硬盘 2 没有创建分区。接下来在 `/dev/sda2` 和 `/dev/sdb` 上创建物理卷, 然后把这两个物理卷加入到逻辑卷组 `vg0` 中, 现在, 逻辑卷组 `vg0` 看起来像一块很大的逻辑硬盘, 然后在其中创建两个逻辑卷 `/dev/vg0/lv0` 和 `/dev/vg0/lv1`。

最后, 分别在 `/dev/sda1`、`/dev/vg0/lv0` 和 `/dev/vg0/lv1` 上创建文件系统, 并分别把它们挂载到文件系统中。在逻辑卷组 `vg0` 中, 还预留有一部分空间未用, 如果在使用中发现某个逻辑卷空间不够用了, 可以在不停机的情况下, 直接调整逻辑卷及其上的文件系统的大小。对服务器来讲, 这是简单但非常有用的功能。

2.3.3 操作系统分区划分样例

表 2-2 是一种供参考的 Linux 操作系统分区划分方式：

表 2-2 Linux 操作系统分区划分方案

分区类型	介 绍	备 注
/boot	启动分区	一般设置为 200MB，boot 目录里包含了操作系统的内核和在启动系统过程中所需要的文件
/	根分区	存放操作系统文件，以及其他一切占用空间不变的文件。有可能的话，尽量扩大该分区的大小。该分区占用量过大极易引发致命的操作系统故障
/home	用户目录	一般每个用户 100MB 左右，可根据操作系统用户的数量来计算相应大小。如果希望在该目录存放大文件，可设为每个用户 1GB 或者 2GB。该分区的大小取决于用户的多少。对于多用户使用的服务器，可以考虑将/home 目录独立挂载到其他的分区中，便于控制用户的权限，比如对用户或者用户组实行磁盘配额限制以及用户权限访问等
/tmp	临时文件	一般设置为 1-5G 左右，方便加载 ISO 镜像文件。对于负载较大的服务器而言，有必要将该目录独立挂载到其他分区，或者酌情扩大该目录所挂载的分区大小，因为该目录是经常引发操作系统故障的文件系统之一，该文件系统容量过低会导致操作系统无法运行
/usr	文件系统	一般设置为 3-15G，绝大多数的用户应用程序的默认安装路径都在这里，就像是 Windows 操作系统的 Windows 目录和 Program Files 目录。与该目录同样著名的是/usr/local 目录，Nagios 的默认安装路径就是/usr/local，因此很多系统管理员都喜欢将/usr/local 目录放到单独的一个逻辑分区里并独立挂载
/var	可变数据目录	包含系统运行时要改变的数据。通常这些数据所在目录的大小是要经常变化的，例如，系统日志是记录在/var/log 下。一般多用户操作系统或者服务器需要建立这个分区，对日志维护很有帮助。一般设置为 2 到 3 个 GB 大小，也可以将剩余的硬盘空间都划分给/var 目录
/opt	附加应用程序目录	存放可选的安装文件，一般可以将一些应用程序的安装包，例如 Java 安装包、Nagios 安装包等资料放在这里



... (faint, mostly illegible text from the background page) ...

第 3 章

配置 VMWARE 虚拟机

本书所指的虚拟机 (Virtual Machine, 或者 VM) 是一种可以在一台物理计算机上模拟出来若干台计算机 (或称逻辑计算机), 每台计算机可以运行单独操作系而互不干扰, 实现一台物理计算机“同时”运行几个操作系统, 还可以将这几个操作系统连成一个网络的软件。

虚拟机体系结构如图 3-1 所示。安装虚拟机的物理计算机成为宿主计算机 (Host PC), 真实的操作系统称为宿主操作系统 (Host OS), 其中安装的虚拟机应用程序可以模拟出一个或多个虚拟机, 在虚拟机运行的操作系统称为客户机操作系统 (Client OS)。虚拟机软件可以在宿主计算机上模拟出来若干台虚拟机, 虚拟机可以同时运行。每个客户机操作系统之间, 以及和主机操作系统之间可以通过虚拟网卡连接成为一个局域网。

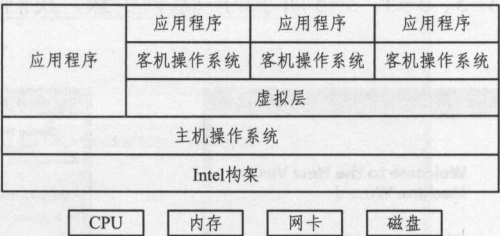


图 3-1 虚拟机架构图

目前, 基于 Intel 处理器的虚拟机典型产品有 VMware (网址: <http://www.vmware.com/>) 的 Workstation、企业级 VSphere, 以及 Microsoft 的 Windows Virtual PC 等。它们均可虚拟出使用 Intel x86 平台的多款 Windows 和 Linux 操作系统, 以及同时运行这些操作系统及应用程序。虚拟机为客户机操作系统提供了一整套虚拟的 Intel x86 兼容硬件, 并虚拟了物理计算机所拥有的全部设备, 包括主板芯片、CPU、内存、SCSI 和 IDE 磁盘设备、各种接口和显示设备等。并且, 每个虚拟机都可以被独立的封装到一个文件中, 可以实现虚拟机的灵活迁移。

虚拟技术从两个方向帮助计算机合理地分配资源，一种是使用虚拟机技术把一个物理的计算机虚拟成若干个独立的逻辑计算机，另一种是使用网络技术把若干个分散的物理计算机虚拟为一个大的逻辑计算机。虚拟机主要采用分区技术，分区能够将物理系统资源划分成多个不同、单独的部分，各部分彼此独立操作，每个分区只能占用一定的系统资源。

在本书中，我们将使用 VMware Workstation 7.1 版本虚拟出一台 Red Hat Enterprise Linux 服务器，通过软件模拟出具备完整硬件系统功能的，运行在一个完全隔离环境中的完整 Nagios 服务器。

3.1 新建虚拟机向导

下面，我们开始安装虚拟机上的操作系统，如图 3-2 所示，是 VMware workstation 的管理界面，选择 New Virtual Machine 选项，可进入虚拟机创建界面。

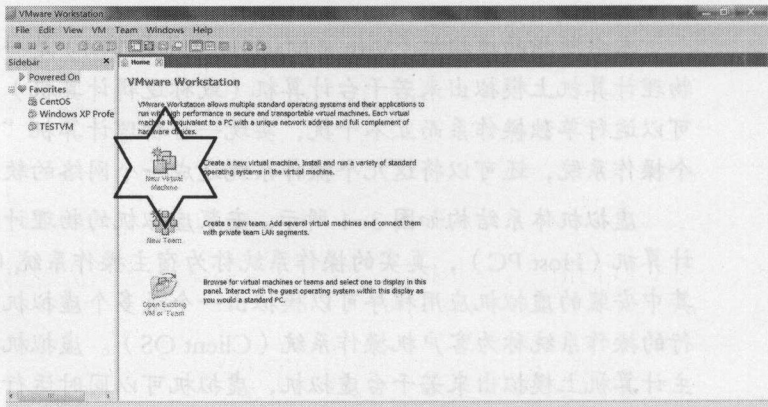


图 3-2 VMware 管理界面

根据如图 3-3、3-4、3-5 所示的一系列步骤，我们可以创建出一个自定义的 Linux 虚拟机清单文件：

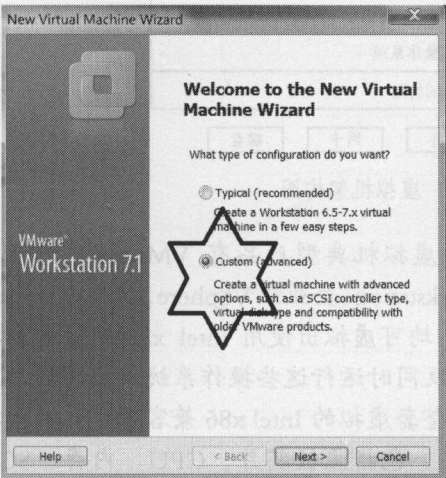


图 3-3 自定义虚拟机创建（高级模式）

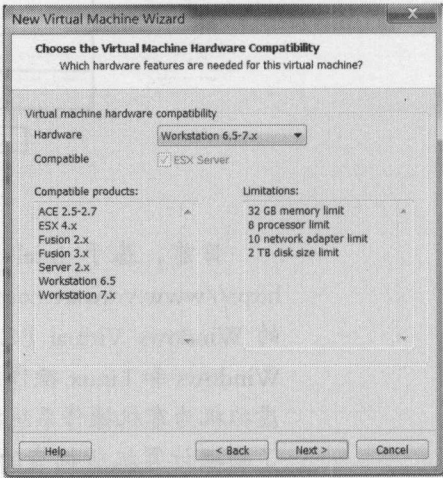


图 3-4 虚拟机硬件兼容模式选择（6.5 及更高兼容模式）

在图 3-6 中，由于没有 Red Hat Enterprise Linux 6 64bit 选项，因此我们选择了与之相近的 Red Hat Enterprise Linux 5 64-bit 选项。

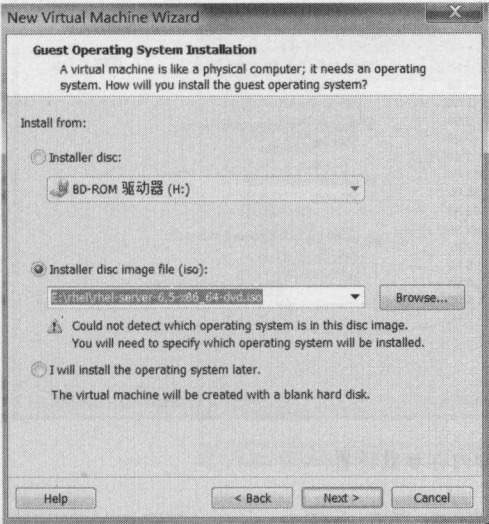


图 3-5 选择操作系统安装文件
(选择之前下载的 ISO 安装文件)

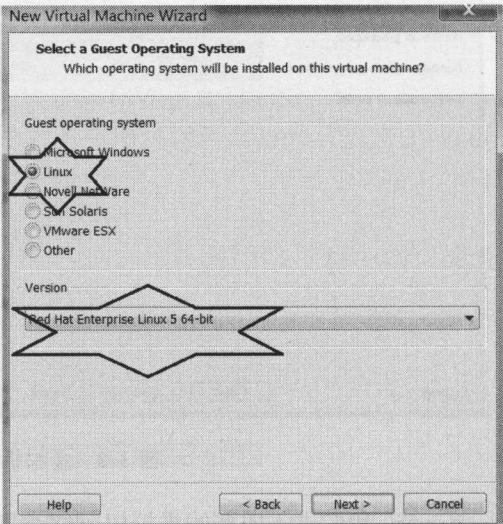


图 3-6 客户机操作系统类型选择

接着在图 3-7 中，我们指定了虚拟机的名称为 rhel_nagios，并将虚拟机文件放在了空间较大的分区里。

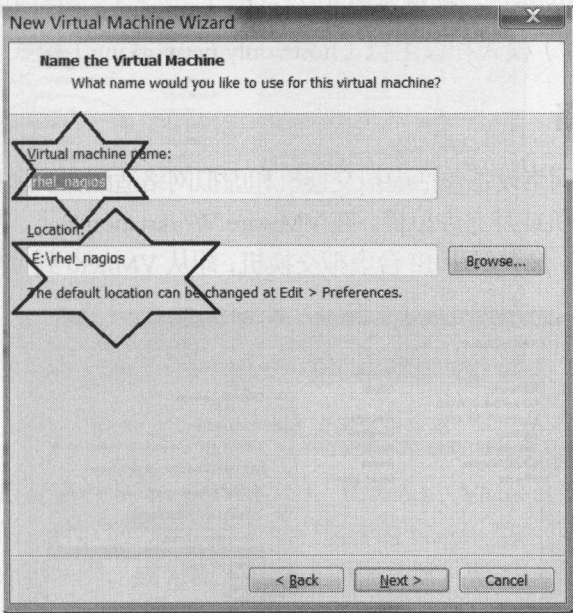


图 3-7 虚拟机名称及文件位置

在接下来的图 3-8 虚拟机 CPU 和内存数量指定页面中，我们分别指定虚拟机 CPU 数量为 1，内存数量为 1024MB，这些都是可根据硬件资源的多少而随时可以调整的参数。

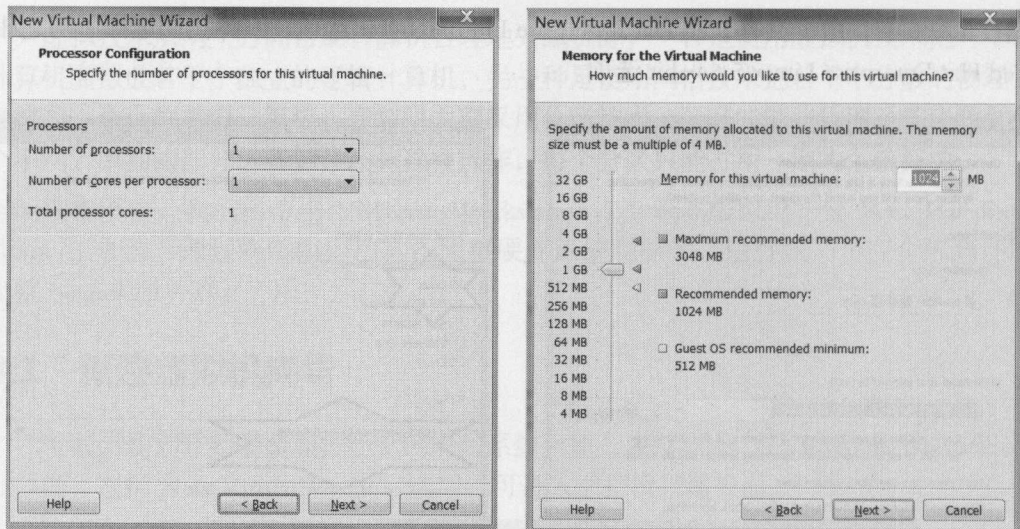


图 3-8 虚拟机 CPU 和内存参数设置

下一步,我们进入重要的虚拟机网络类型设置页面,首先介绍 VMware 虚拟机的联网模式。

3.2 VMware 的联网模式简介

VMware 的难能可贵之处在于,它不但能够虚拟出单一的系统,而且能够虚拟出复杂的网络。在这样的网络中,除了需要了解虚拟网络设备以及虚拟网络服务外,还要知道 VMware 所提供的 3 种虚拟机联网模式,即桥接模式 (bridged networking)、网络地址转换 (network address translation, NAT) 模式和仅主机 (host-only networking) 模式。

3.2.1 虚拟网络设备

VMware 所虚拟的网络设备包括虚拟交换机和虚拟网络适配器。虚拟交换机能将一台或多台虚拟机连接到宿主主机或其它虚拟机。在 VMware Workstation 7 下,可以根据组网的需要虚拟出一定数量的交换机,最多可用 10 台虚拟交换机,即从 VMnet0 到 VMnet9,如图 3-9 所示。

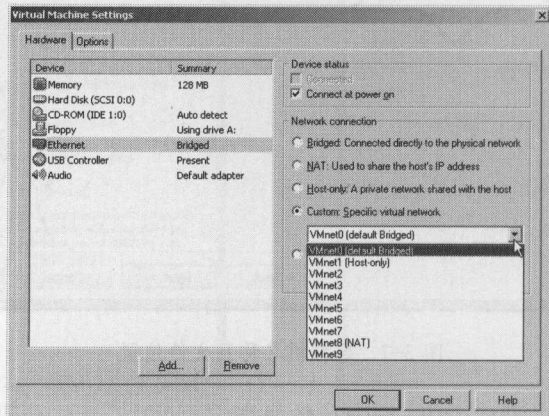


图 3-9 虚拟交换机

在新建虚拟机过程中,无论选择了桥接模式、仅主机模式和 NAT 模式任一种连网模式,都会为虚拟机自动创建虚拟网络适配器,即 Virtual Ethernet Adapter,也称为虚拟网卡,如图 3-10 所示。

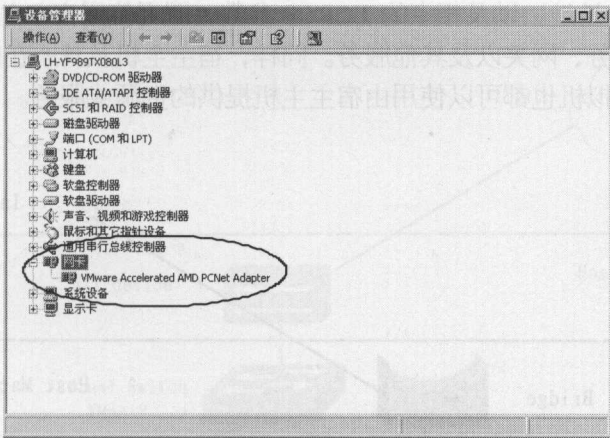


图3-10 虚拟网络适配器

此外,还能够通过虚拟机配置面板,为一个虚拟机最多设置 3 个虚拟网络适配器,如图 3-11 所示。设置完毕后进入客户机操作系统,就可以在设备管理器中发现这三个虚拟网络适配器。

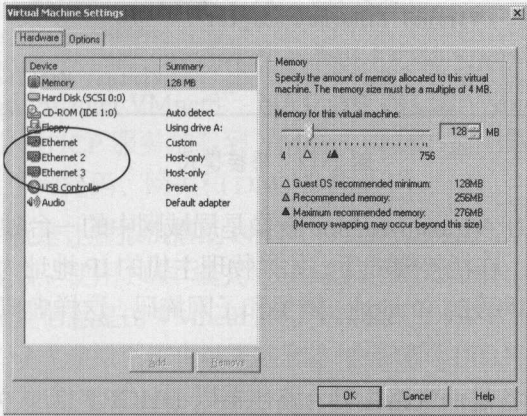


图 3-11 创建多个虚拟网络适配器

为虚拟机创建了虚拟交换机 VMnet (VMnet0、VMnet1、VMnet8 等等)和虚拟网卡后,下一步需要考虑的是采用何种联网方式。由于我们创建的 Nagios 监控服务器必须依靠宿主机才能存活,并且需要与外界联网,成为真正意义上的服务器。要想达成目标就必须根据实际情况选择 3 种联网方式中的一种,下面将对这 3 种联网方式进行简单介绍。

3.2.2 虚拟机联网方式之桥接模式 (bridged networking)

图 3-12 是采用桥接模式连网的示意图。其中,虚拟网桥 (Virtual Bridge) 通过连接宿主

主机（Host Machine）中的物理以太网适配器和虚拟机中的以太网适配器（Virtual Ethernet Adapter），将虚拟机连接到宿主主机所在的局域网（或者 Internet）。默认情况下，虚拟网桥使用虚拟交换机（Virtual Switch）VMnet0 的虚拟网络。对于桥接的虚拟机，只要配置与宿主主机同一网段的 IP 地址，以及相应的 TCP/IP 参数，即可使用宿主主机所有的网络资源，包括打印机、文件服务、网关以及其他服务。同样，宿主主机及其所在的网络上的任何物理计算机，连同其他虚拟机也都可以使用由宿主主机提供的资源或服务。

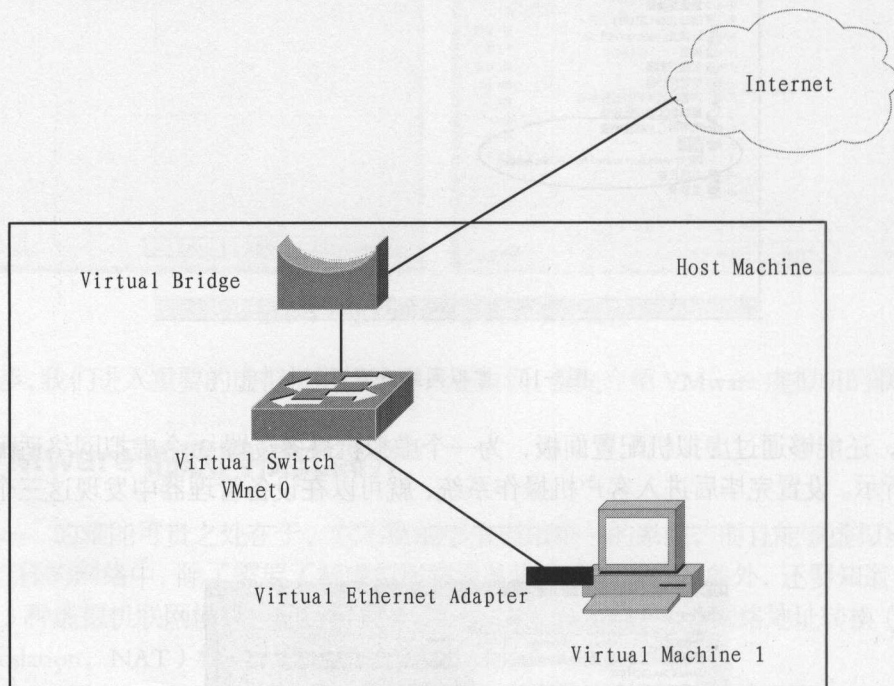


图 3-12 桥接模式

在桥接模式下，虚拟机 Virtual Machine 就像是局域网中的一台独立的物理主机，它可以访问网内任何一台服务器。在桥接模式下，如果物理主机的 IP 地址为手工配置，那么同样需要为虚拟机手工配置同一网段的 IP 地址、网关和子网掩码，这样虚拟机才能和宿主主机进行通信，并连接到 Internet。

如果宿主主机的上联设备是一台路由器，路由器以 DHCP 的方式为宿主主机分配 IP 地址，那么该宿主主机上的虚拟机 IP 可以由同一台路由器以 DHCP 的方式分配，以实现通过局域网的网关或路由器访问互联网。

3.2.3 虚拟机联网方式之网络地址转换（network address translation , NAT）

模式

图 3-13 使用 NAT 模式连网。其中，NAT 设备（NAT device）通过连接宿主主机（Host Machine）中的物理以太网适配器和虚拟机中的以太网适配器，将虚拟机（Virtual Machine）连接到宿主主机所在的局域网（或 Internet）。但与桥接模式不同的是，虚拟主机与宿主主机

并非处于同一网段，而是处于与宿主主机相隔离的私有网段。在这个私有网段中，虚拟主机通过虚拟 DHCP 服务器得到 IP 地址。NAT 设备（NAT device）对该 IP 地址与宿主主机连接到外部网络的 IP 地址进行相互转换，保证虚拟机能够单向访问外部网络，但外部网络不能访问到虚拟机及所在的私有网络。

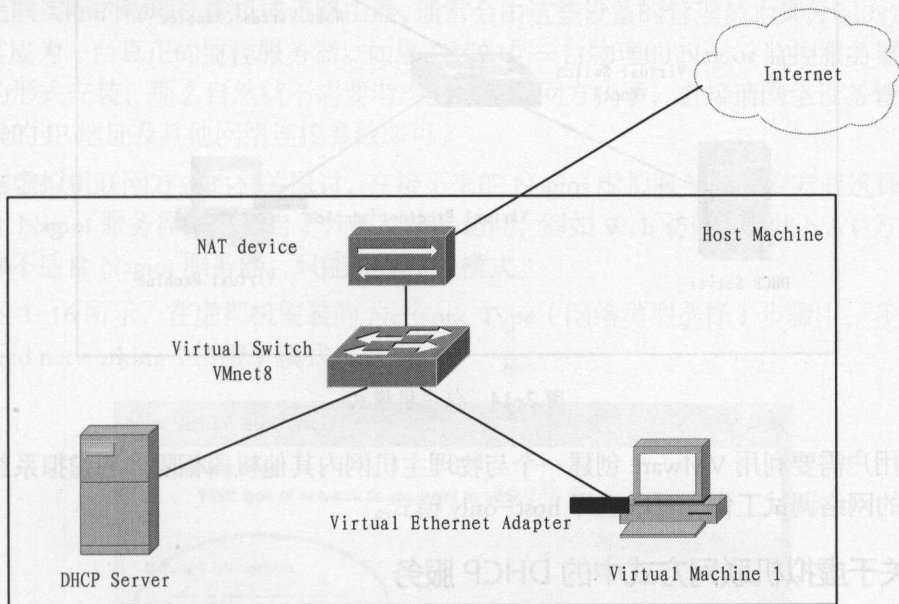


图 3-13 NAT 模式

VMware Workstation 安装时，自动在宿主主机上安装 NAT 设备，可以在设备管理器中发现“VMware Network Adapter VMnet8” NAT 适配器（即 VMnet8，默认的 NAT 适配器）。在 NAT 模式下，会由 DHCP 服务加载到 VMnet8（NAT 适配器）上，并由 DHCP 服务为虚拟机提供 IP 地址、子网掩码、网关和 DNS 参数。

使用 NAT 模式，就是让虚拟机借助 NAT（网络地址转换）功能，通过宿主机所在的网络来访问公网。也就是说，使用 NAT 模式可以实现在虚拟系统里访问互联网。NAT 模式下的虚拟主机的 TCP/IP 配置信息是由 VMnet8 虚拟交换机的 DHCP 服务器提供的，无法进行手工修改，因此使用 NAT 模式虚拟系统也就无法和本地局域网中的其他真实主机进行通讯。

用 NAT 模式最大的优势是虚拟系统接入互联网非常简单，你不需要进行任何其他的配置，只需要宿主机能访问互联网，虚拟机就能访问互联网。

3.2.4 虚拟机联网方式之仅主机（host-only networking）模式

在某些特殊的网络调试环境中，要求将物理机所在的真实环境和虚拟机环境隔离开，这时你就可采用仅主机模式（host-only networking）。在该模式中，物理机所属的所有虚拟机是可以相互之间通信的，但虚拟机和物理机所在的真实网络（或者 Internet）是被隔离开的，虚拟机不能访问互联网。

图 3-14 使用仅主机模式联网，该模式创建的虚拟机网络仅在宿主主机内部，虚拟机通过虚拟以太网适配器和宿主主机通信。仅主机模式只能使用私有 IP，在仅主机模式下，会由

DHCP 服务加载到 VMnet1（host-only 模式默认的虚拟交换机）上，并由该虚拟交换机上的 DHCP Server 为虚拟机提供 IP 地址、子网掩码、网关和 DNS 参数。

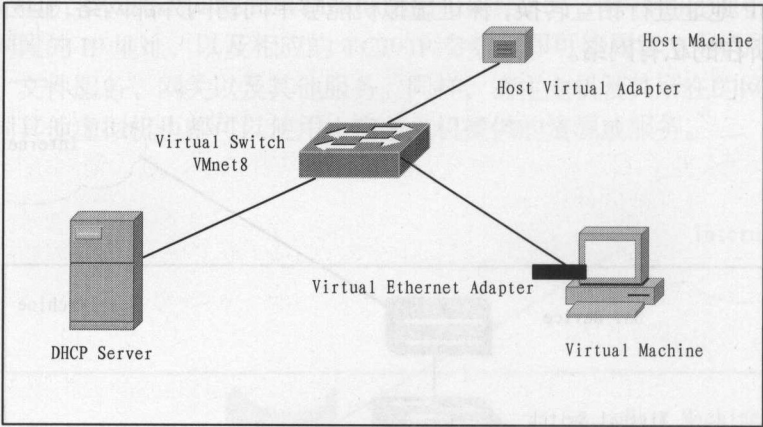


图 3-14 仅主机模式

如果用户需要利用 VMware 创建一个与物理主机网内其他机器相隔离的虚拟系统，进行某些特殊的网络调试工作，可以选择 host-only 模式。

3.2.5 关于虚拟机联网方式中的 DHCP 服务

VMware Workstation 安装时，宿主主机（即物理主机）自动安装 VMware DHCP（动态主机配置协议）服务，如图 3-15 所示。这样，将宿主主机变成了一台 DHCP 服务器，为使用 NAT 模式和仅主机模式的虚拟机自动分配动态 IP 地址、子网掩码、网关和 DNS 参数。

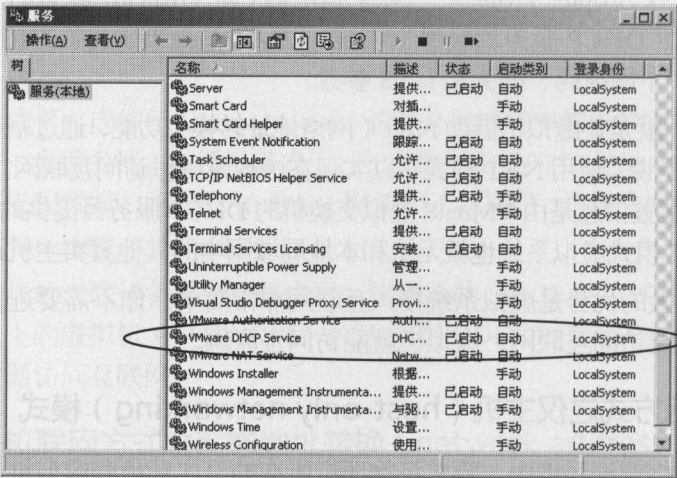


图 3-15 虚拟机联网方式中的 DHCP 服务

此处需要着重强调的是，如果虚拟机采用桥接方式联网，那么将不会以 DHCP 的方式从宿主主机（即物理机）获取 IP，必须由用户手工指定该虚拟机的 IP 地址、子网掩码、网关等网络参数，或者以 DHCP 的方式从上联的物理网络设备（例如具备 DHCP 服务的物理交

换机或者物理路由器)获取 IP 地址后,方可接入物理机所在的外部网络(含 Internet)。

3.2.6 选择 Nagios 虚拟服务器的联网方式

在实际的工作中,如果用户想安装一台能够投入运行的虚拟 Nagios 服务器,那么虚拟机就需要上联实际的物理交换机或者路由器,通常会由这些设备的管理员为其分配固定的 IP 地址,使其成为一台真正的监控服务器。如果安装的是一台物理的 Nagios 监控服务器,而非以虚拟机的形式安装,那么自然就不需要考虑虚拟机联网方式了,直接请网络设备管理员为其配置正确的 IP 地址及其他网络连接参数即可。

根据虚拟机联网方式的有关探讨,在接下来的 Nagios 虚拟服务器联网方式选择中,由于我们要让 Nagios 服务器接受来自于外部网络的访问,例如 Web 访问,因此 NAT 方式和仅主机方式都不适合 Nagios 服务器,只能选择桥接模式。

如图 3-16 所示,在虚拟机安装的 Network Type (网络类型选择)步骤中,我们选择了 Use bridged networking (桥接)模式。

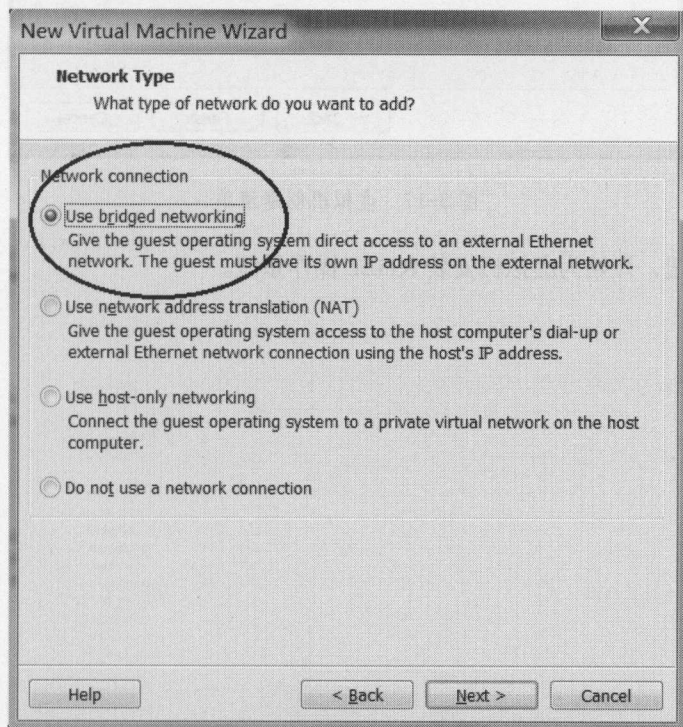


图 3-16 虚拟机安装联网方式选择

3.3 完成虚拟机创建向导并查看配置清单

在接下来的虚拟机创建选项中,我们选择默认选项即可。最后,我们得到了一张自定义的虚拟机配置清单,如图 3-17 所示。

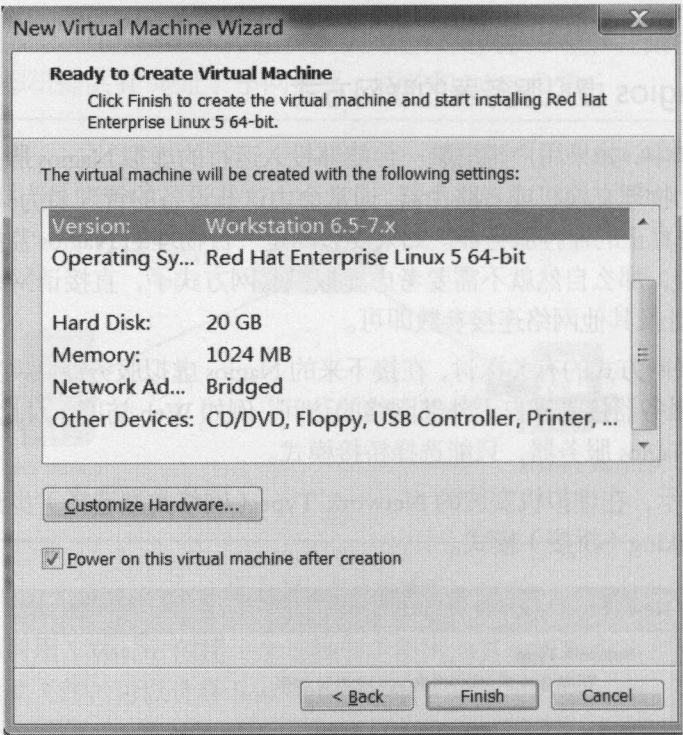


图 3-17 虚拟机创建清单

单击 Finish 按钮，开始为虚拟机安装 RHEL 操作系统。

第4章

为虚拟机安装 RHEL 操作系统

RHEL, 即 Red Hat Enterprise Linux 的缩写, 是 Red Hat 公司的 Linux 系统。

4.1 引导菜单

图 4-1 是操作系统安装引导界面，选择第一项，即 Install or upgrade an existing system。

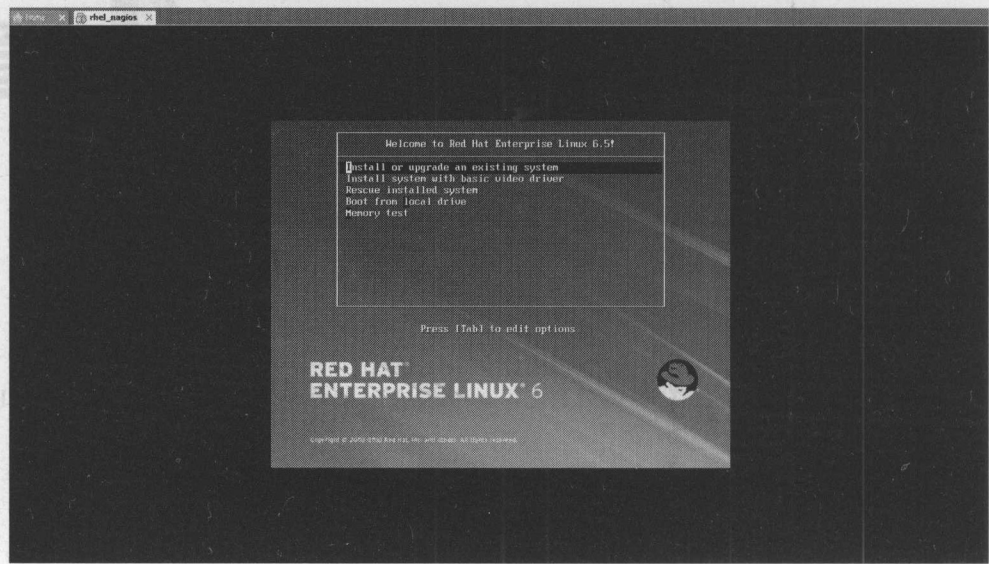


图 4-1 操作系统安装引导菜单

图 4-1 所示引导菜单选项如下：

- 安装或升级现有系统 (Install or upgrade an existing system)：这个选项是默认的。选择此选项，将进入操作系统的图形安装界面。
- 使用基本的视频驱动程序安装系统 (Install system with basic video driver)：此选项允许用户在程序无法正常加载视频卡的条件下安装。如果选择第一项 “Install or upgrade an existing system” 时，屏幕上出现扭曲或一片空白，重新启动计算机，并尝试此选项。
- 救援已安装的系统 (Rescue installed system)：如果已经安装的操作系统在启动时发生问题，选择这个选项来修复已经安装的操作系统的。虽然 RedHat Enterprise Linux 是一个非常稳定的计算平台，仍有可能发生偶然的问题而无法正常工作。救援环境包含的实用程序能够帮忙解决这些问题。
- 从本地驱动器启动 (Boot from local drive)：此选项从磁盘正常引导已经安装的操作系统的，而非从光盘引导。
- 内存检测 (Memory test)：执行安装前的内存检测。

4.2 操作系统安装欢迎界面 (语言及键盘布局)

在一阵软硬件初始化操作过后，进入如下的检测安装介质页面，单击 Skip 按钮跳过该页面，进入图 4-2 中，熟悉的带有 Red Hat 图标的操作系统安装欢迎界面。

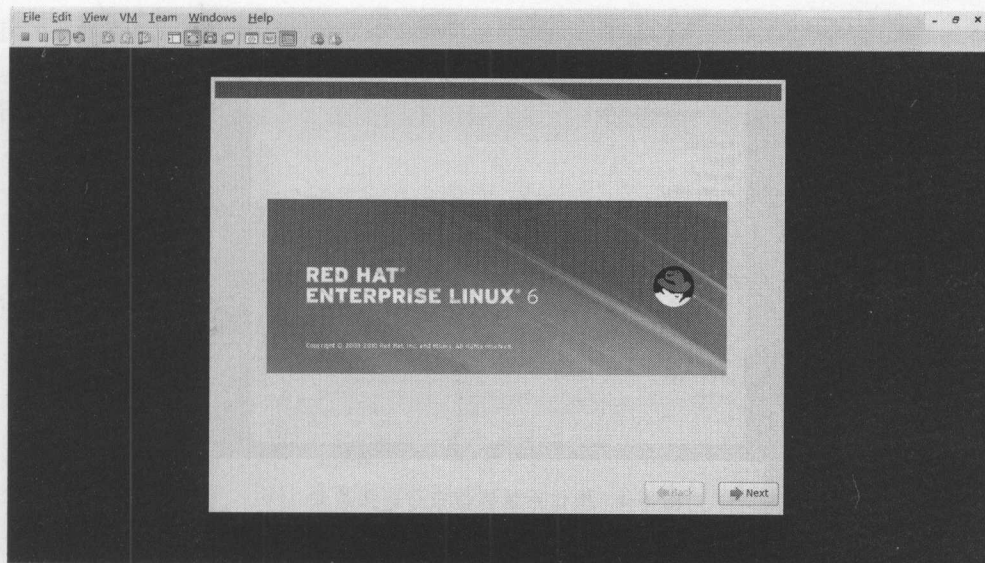


图 4-2 带有 Red Hat 图标的安装界面

单击右下角的 Next 按钮，进入安装过程语言选择菜单，为了便于日后的管理，我们选择“中文（简体）”，使操作系统的默认语言为中文，如图 4-3 所示。

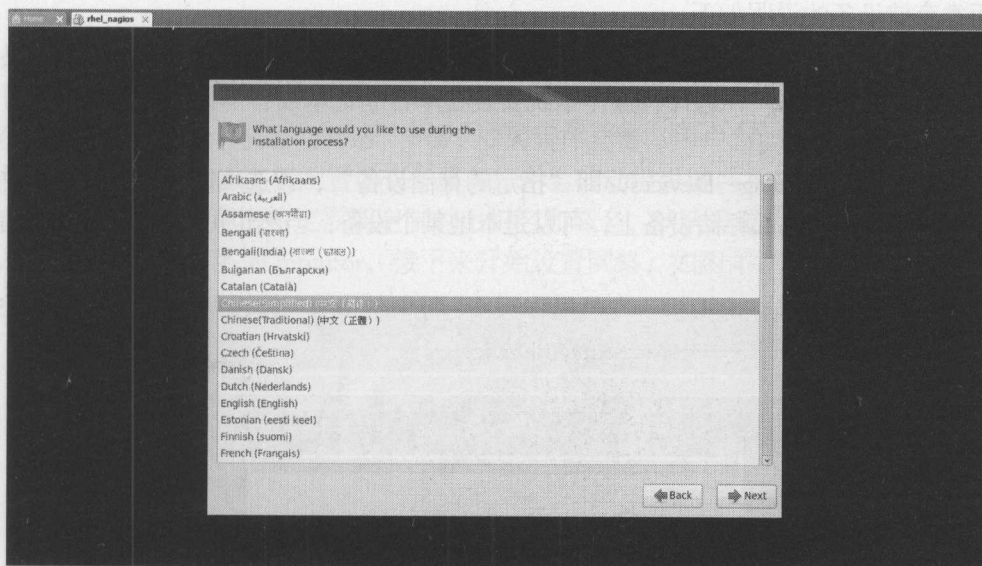


图 4-3 操作系统语言选择菜单

继续单击 Next 按钮，进入操作系统键盘布局类型选择界面。选择键盘类型一般默认会选择“美国英语式（U.S.English）”，即美式键盘，在此使用默认的选择，如图 4-4 所示。

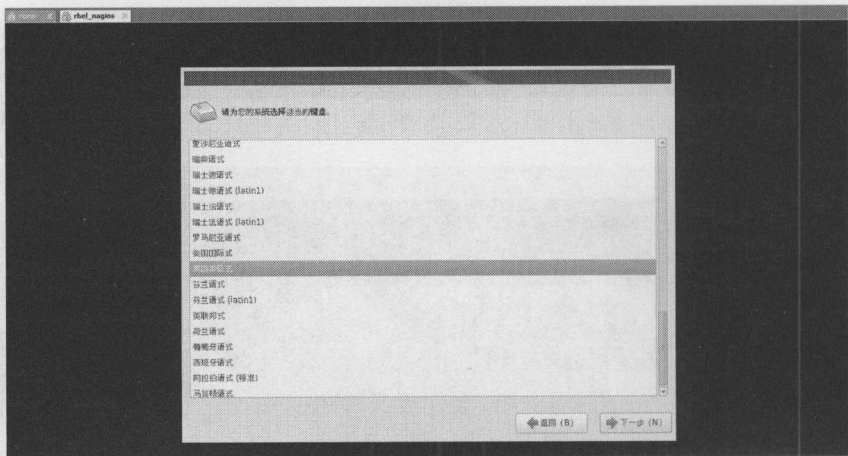


图 4-4 操作系统键盘布局选择菜单

4.3 存储设备选择

下面是存储设备选择页面。为了将操作系统安装在计算机上，你需要选择从两类存储设备中选择一种。

两类存储设备的说明如下：

- Basic Storage Devices，即“基本存储设备”，是操作系统默认安装地点。如果您是初次安装的用户，选择将操作系统安装在单台笔记本或者单台个人电脑上，可以选择选择该选项。
- Specialized Storage Devices，即“指定的存储设备”，该选项需要用户将系统安装指定到特定的外部存储设备上，可以是本地某个设备，当然也可以是 SAN（存储局域网）。

由于我们这次选择在虚拟机上安装操作系统，而非在外接存储设备上，因此在图 4-5 中，我们选择默认的安装选项——Basic Storage Devices。

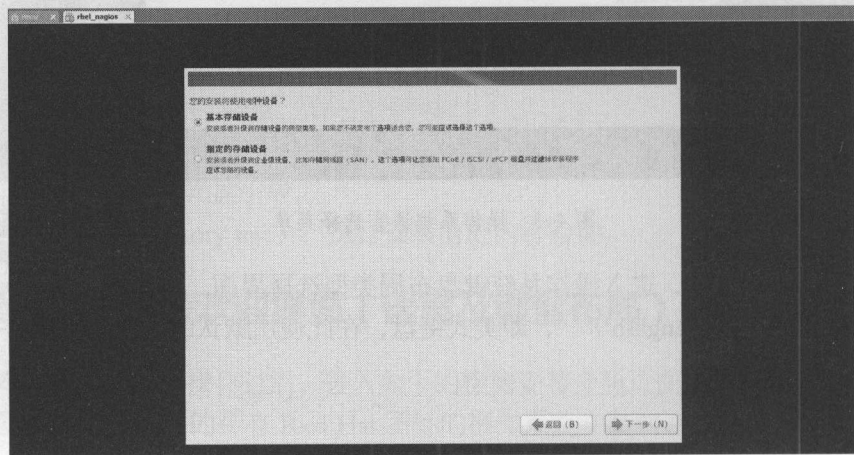


图 4-5 存储设备选择

接下来是初始化硬盘的相关警告。如果硬盘上没有找到分区表，安装程序会要求初始化硬盘。此操作使硬盘上的任何现有数据无法读取。由于我们这次是全新安装，所以单击第一个按钮“是，忽略所有数据”，忽略掉虚拟磁盘上已有的数据，对该虚拟磁盘执行初始化，如图 4-6 所示。

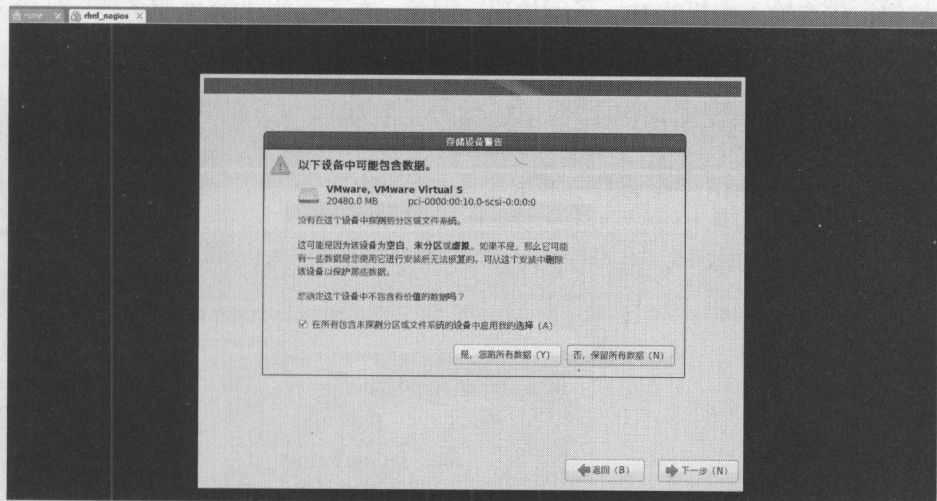


图 4-6 忽略磁盘数据

4.4 主机名与网络设置

在接下来的“设置主机名与网络”步骤，安装程序会提示用户为主机设置主机名和域名。许多网络有 DHCP（动态主机配置协议）服务，它会自动提供域名系统的一个连接，让用户输入一个主机名。除非存在特定需要定制的主机名和域名，默认设置为 localhost.localdomain。在此，我们将虚拟机命名为 monitor，接下来开始设置网络，如图 4-7 所示。

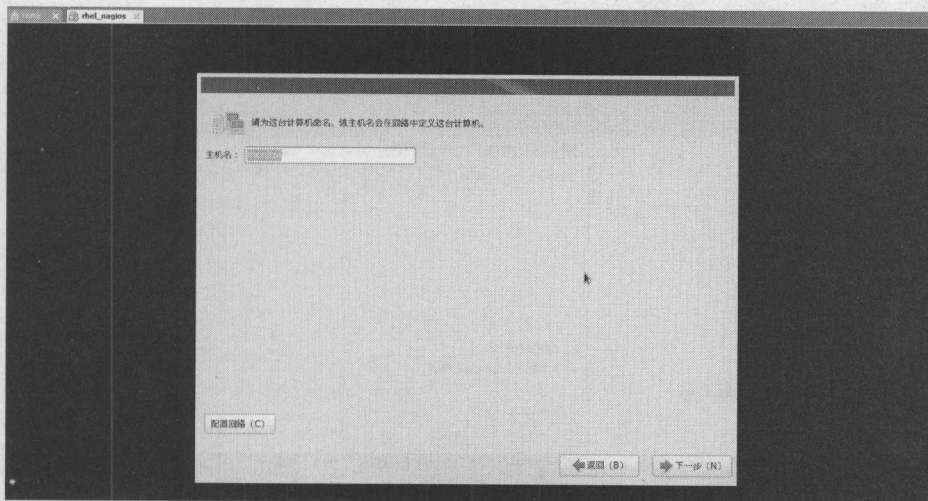


图 4-7 服务器命名

在“设置固定 IP”一步中，如果规划的虚拟机具有独立的固定 IP（参照章节0），那么可以按照下列步骤为虚拟机设定固定 IP。

选择“配置网络”→“有线”→“System eth0（虚拟机默认网卡）”→“编辑”，在弹出编辑窗口上选择“IPv4 设置”，打开“方法”边上的下拉菜单，选择“手动”选项。单击“添加”按钮，依次输入本机的 IP、子网掩码、网关。在下面的“DNS 服务器”处输入 DNS 地址。最后，单击“应用”按钮即可，如图 4-8 所示。

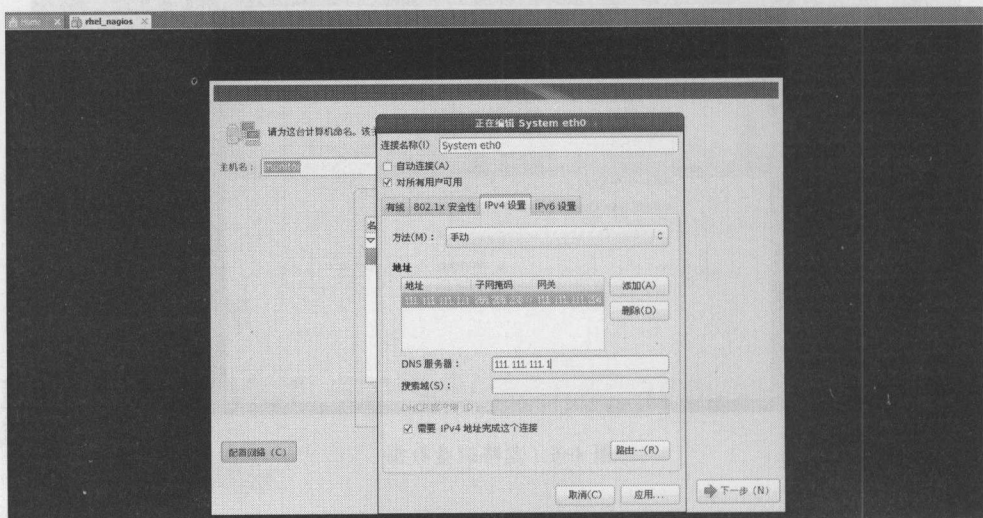


图 4-8 手工指定 IP 地址

如果虚拟机使用 DHCP 的方式从上联物理交换机或者物理路由器上获取 IP 地址，那么可以保持该项不变，如图 4-9 所示。

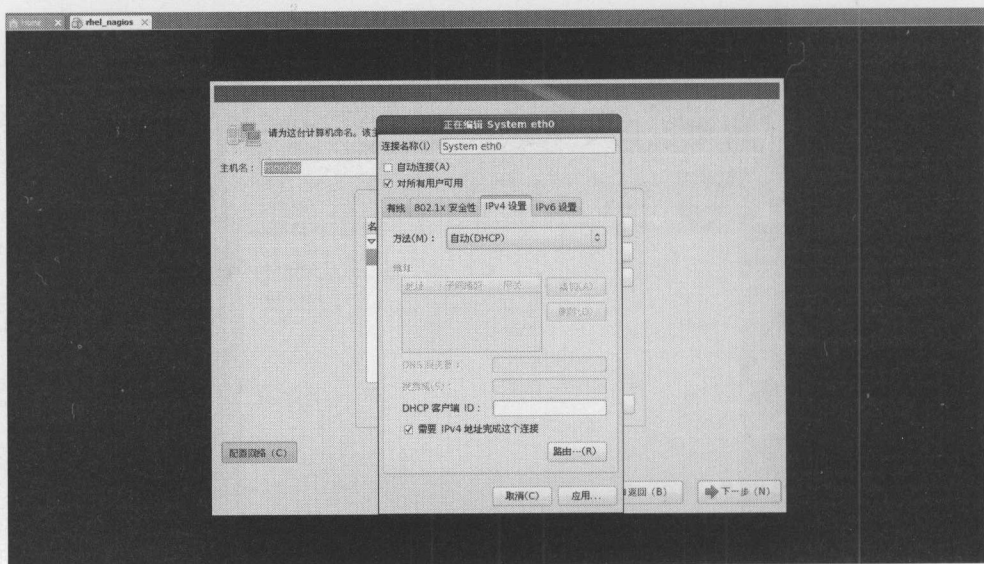


图 4-9 DHCP 方式获取 IP 地址

在本次安装过程中，由于我们在 3.2.6 小节中选择了使用“桥接模式”联网，且物理机上联的路由器有 DHCP 服务，故我们只需保持“配置网络”中的参数不变即可。

4.5 时区选择

因为全世界分为 24 个时区，所以，要告知系统时区在哪里。如下图所示，用户可以选择北京，或直接用鼠标在地图上选择。要特别注意 UTC，它与“夏令时”有关，一般不需要选择这个选项，否则会造成时区混乱，导致系统显示的时间与本地时间不同。

在图 4-10 时区选择页面中，选择“亚洲/上海”，不选择“系统时钟使用 UTC 时间”。

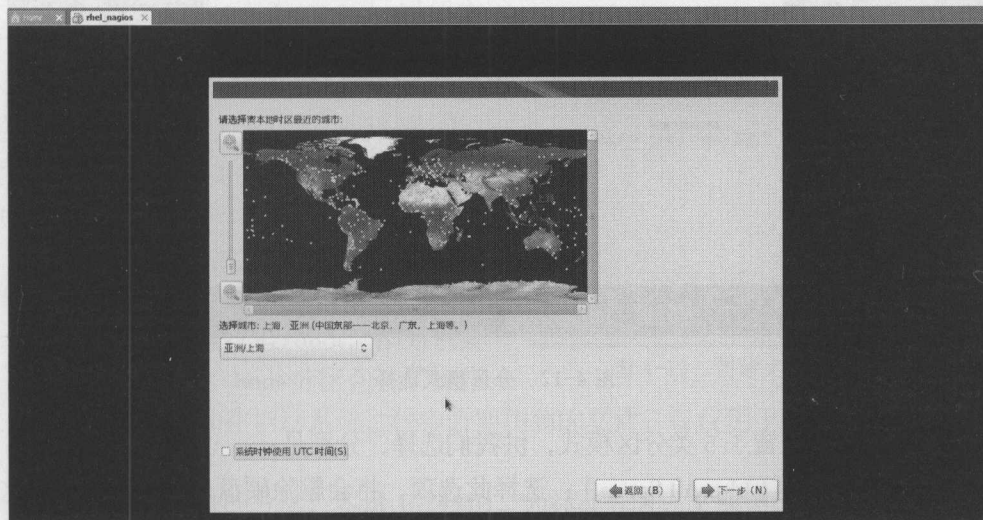


图 4-10 时区选择

在图 4-11 密码填写页面中，我们需要为操作系统指定一个具备复杂度的密码。



图 4-11 密码填写页面

4.6 磁盘分区设置

下面进入重要的磁盘分区设置页面，如图 4-12 所示。

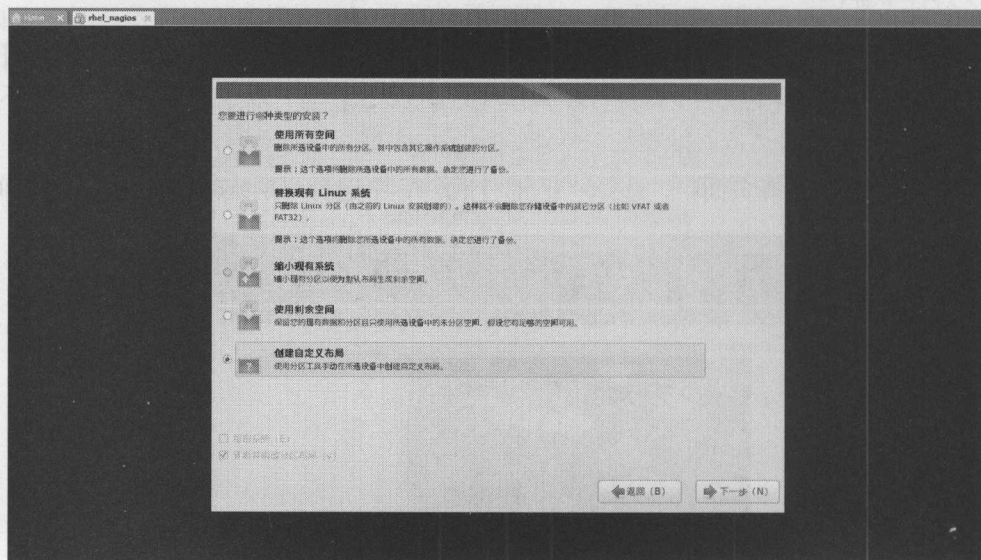


图 4-12 分区模式选择

Red Hat 为我们预置了 5 类分区模式，供我们选择，分别是：

- 使用所有空间（Use All Space）：选择此选项，将会删除硬盘上的所有分区（这包括如 Windows 的 FAT 分区、NTFS 分区或其他操作系统创建的分区），整块硬盘上的数据将被清零，包括分区上安装的操作系统和应用软件，然后重新安装 Linux 操作系统。因此选择此项时一定要慎重，只要当确保该硬盘上的所有数据都无效时，才可以选择此项。
- 替换现有 Linux 系统（Replace Existing Linux System）：选择此选项，将会删除先前安装 Linux 时所创建的分区，并且不会删除其他分区（如 Windows 操作系统的 FAT 分区或 NTFS 分区等）。例如，如果用户之前在硬盘上既安装了 Linux 操作系统，又安装了 Windows 操作系统，在本次安装中希望在重新安装 Linux 操作系统的同时保留 Windows 操作系统，那么可以选择该选项。相反地，如果硬盘上有想保留的 Linux 操作系统，请不要选择该选项。
- 缩小现有系统（Shrink Current System）：如果用户的整个硬盘已被一个操作系统分区（例如，Windows 操作系统的 NTFS 或 FAT 分区）占用了，就会看到这个选项。该选项会在不清除原有分区上资料（例如，不影响原有 Windows 操作系统运行）的情况下缩小分区，并在腾出的空间上安装 Linux 操作系统。
- 使用剩余空间（Use Free Space）：该选项不会删除任何分区，无论是之前安装过 Windows 操作系统，还是其他类型的 Linux 操作系统。只会在尚未分给任何操作系统的剩余空间上进行自动分区。如果用户的硬盘早已被另一个操作系统占满了，此项便无法选择。

- 创建自定义布局（Create Custom Layout）：选择此选项，在该硬盘上手动进行分区并创建自定义的文件系统布局。

在第 23 页的表 2-2 Linux 操作系统分区划分方案中，我们已经规划了虚拟 Nagios 服务器的 Linux 文件系统划分方式，下面就按照“创建自定义布局”方式来进行正式的文件系统划分。

4.7 划分文件系统

如图 4-12 所示，选择“创建自定义布局”并单击“下一步”按钮，出现如图 4-13 所示界面：

在图 4-13 中，显示出系统识别到的唯一一块硬盘，也就是我们为虚拟机所分配的 20G 磁盘空间。根据表 2-2 中的规划，我们确定了如下的磁盘分区策略：

- 分出一个单独的分区，为标准分区，文件类型为 ext4，容量 100MB，用来挂载/boot，作为系统的 boot 分区。boot 分区包含了操作系统的内核和在启动过程中所要用到文件，仅仅用于开机引导程序，而没有其他用途。
- 分出一个单独的分区，容量 1024MB，为标准分区，文件类型为 ext4，用来挂载/，作为系统的根分区，该分区是安装操作系统软件的地方。
- 分出一个单独的分区，容量 1024MB，用来挂 swap 分区，作为系统的交换分区。
- 最后将剩余空间分出一个较大分区，转换为 PV，在该 PV 基础上创建 VG，命名为 filevg_monitor，最后将 PV 加入该 VG。

接下来，在图 4-13 的基础上，选中图中的“空闲”选项，并单击“创建”按钮，然后按照如图 4-14~图 4-21 所示进行设置。

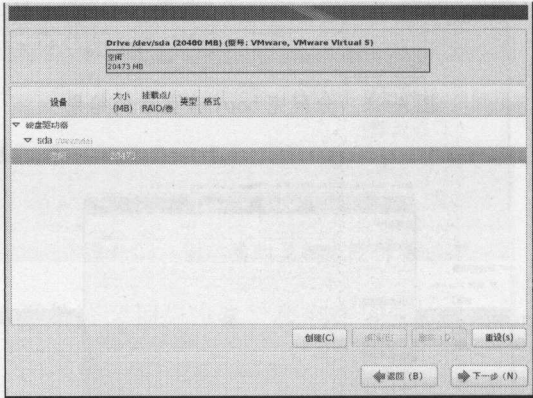


图 4-13 创建文件系统之选择硬盘

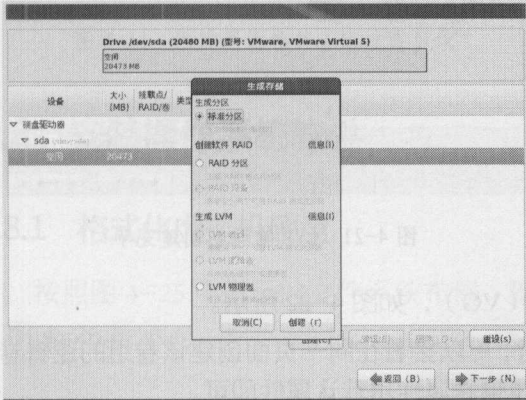


图 4-14 创建 boot 分区

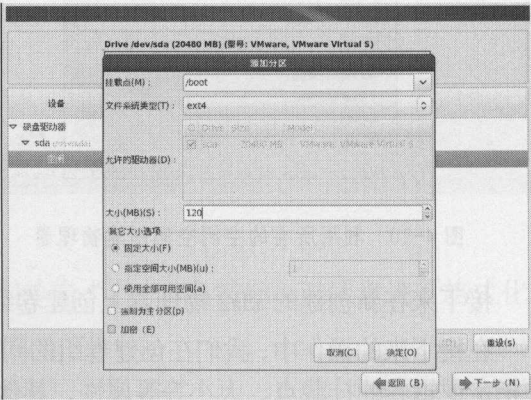


图 4-15 分配 boot 分区空间

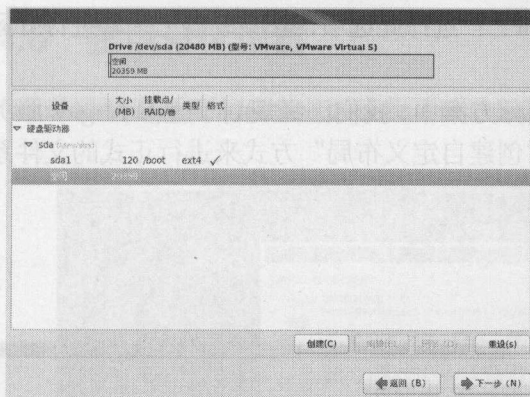


图 4-16 分配完 boot 分区后的结果

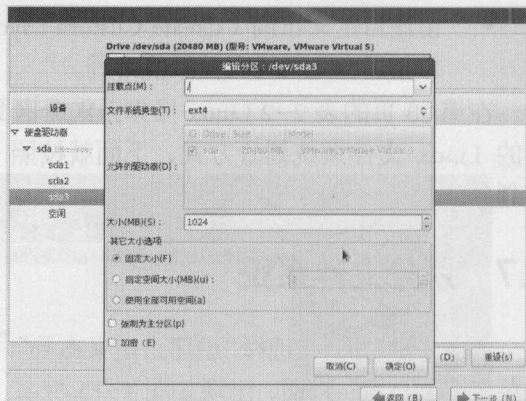


图 4-17 创建根分区

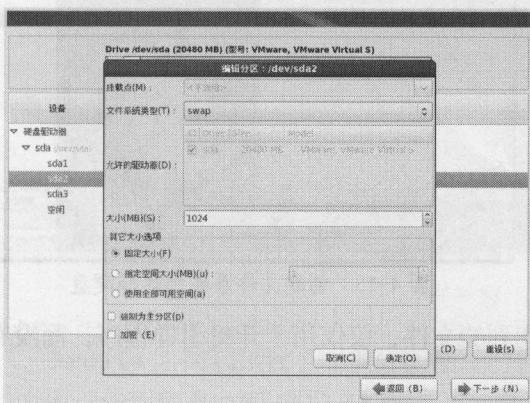


图 4-18 创建交换分区

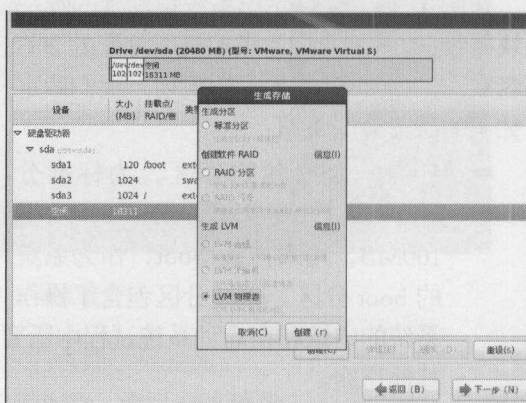


图 4-19 在空闲空间上创建 LVM 物理卷

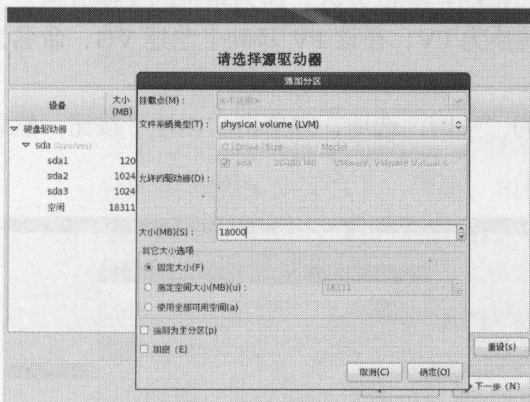


图 4-20 利用所有的空闲空间创建物理卷

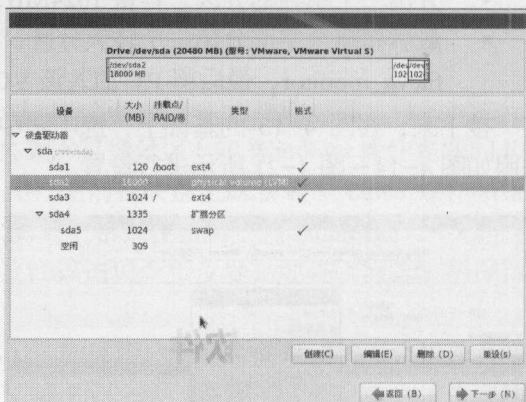


图 4-21 LVM 物理卷创建完毕

接下来在新创建的 sda2 物理卷上创建卷组 (VG)，如图 4-22 所示。

在接下来的操作中，我们在创建卷组的同时，可以接着在同一页面创建该卷组的逻辑卷，并指定逻辑卷的挂载点、大小等等属性，其他逻辑卷属性用默认属性即可。

在图 4-23 中，我们指定创建的 LVM 卷组名为 filevg_monitor，接着单击左下角的“添加”按钮，可以在该卷组上创建逻辑卷 LogVol00，指定该逻辑卷挂载点为 /home 目录，容量为 2048MB。

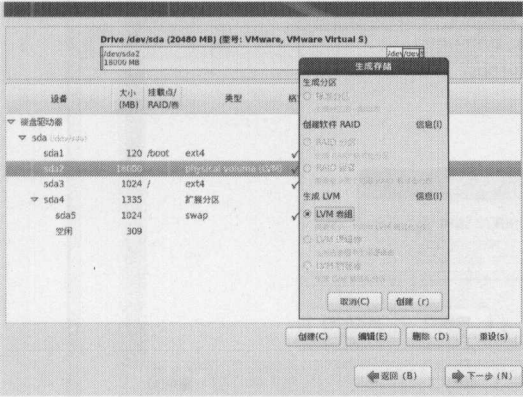


图 4-22 创建卷组

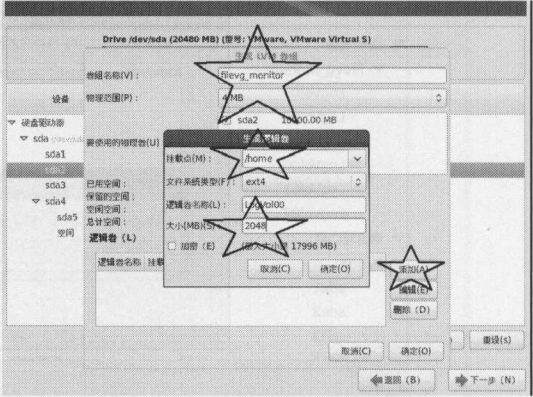


图 4-23 创建卷组及相应逻辑卷

如图 4-24，按照第 23 页表 2-2 Linux 操作系统分区划分方案，我们在 filevg_monitor 上连续创建了 /opt, /var, /usr 和 /usr/local 目录。

最终的分区方案如图 4-25 所示。

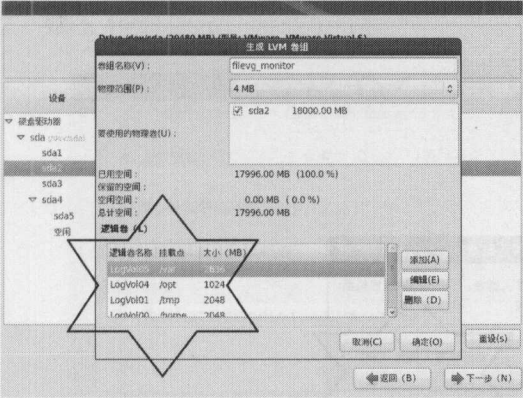


图 4-24 创建逻辑卷并指定挂载目录

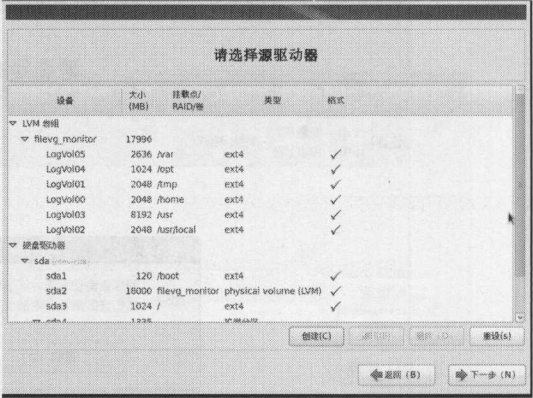


图 4-25 最终的文件系统布局

4.8 安装操作系统软件

4.8.1 格式化虚拟机硬盘

按照图 4-25 中所示的文件系统布局，我们单击“下一步”按钮，对硬盘进行格式化，如图 4-26 所示。

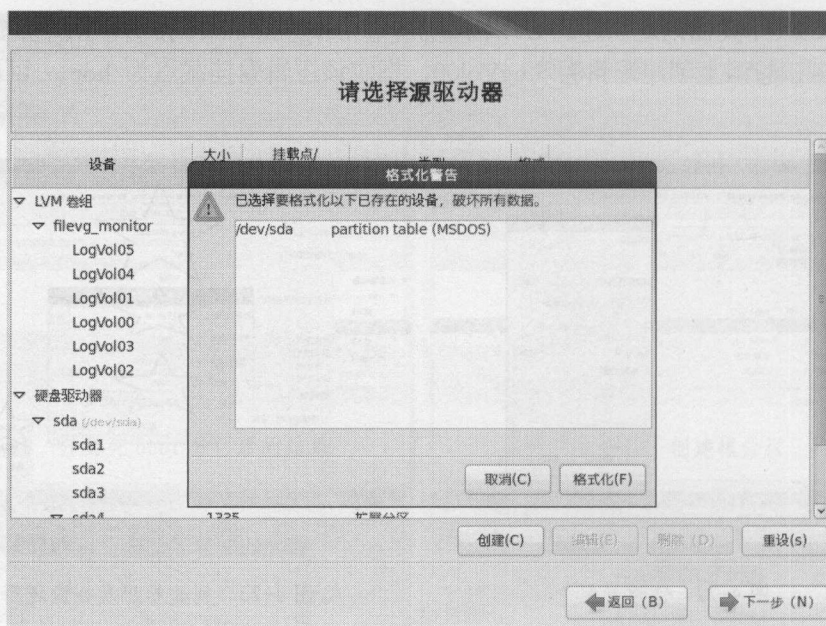


图 4-26 格式化硬盘

单击“将修改写入磁盘”按钮，如图 4-27 所示。

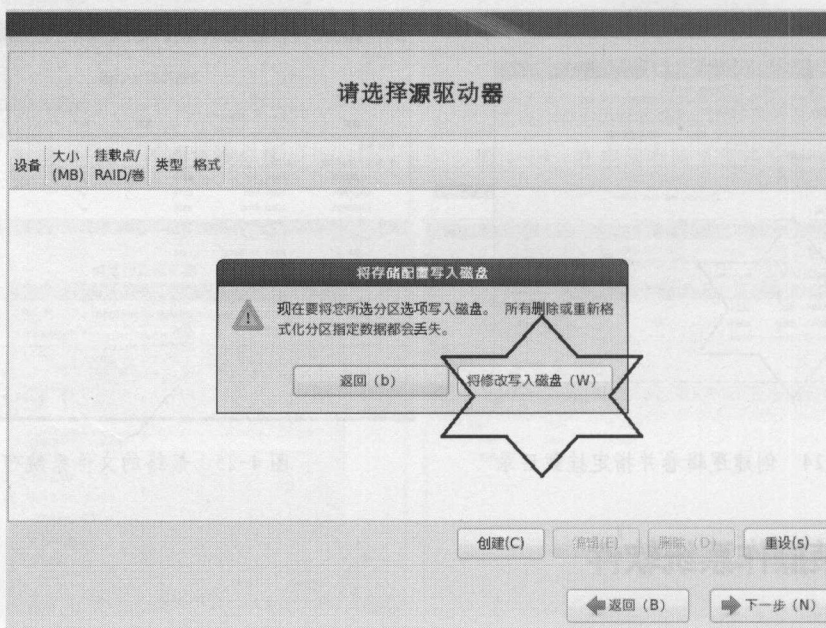


图 4-27 将分区信息写入磁盘

如图 4-28 所示，系统格式化磁盘并创建文件系统。

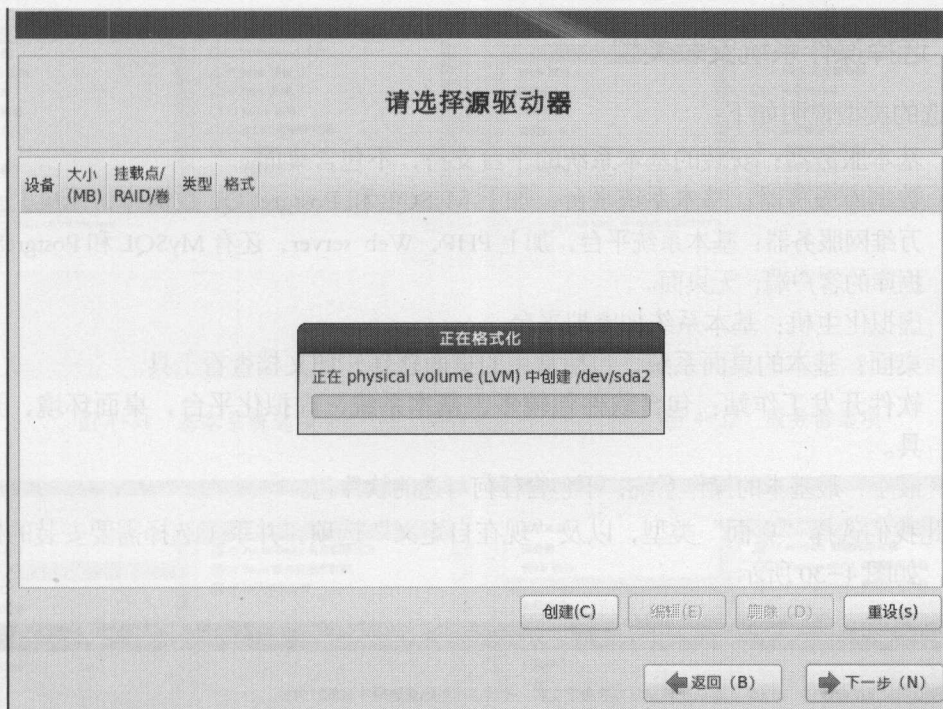


图 4-28 格式化磁盘并创建文件系统

如图 4-29 所示，装载引导程序。

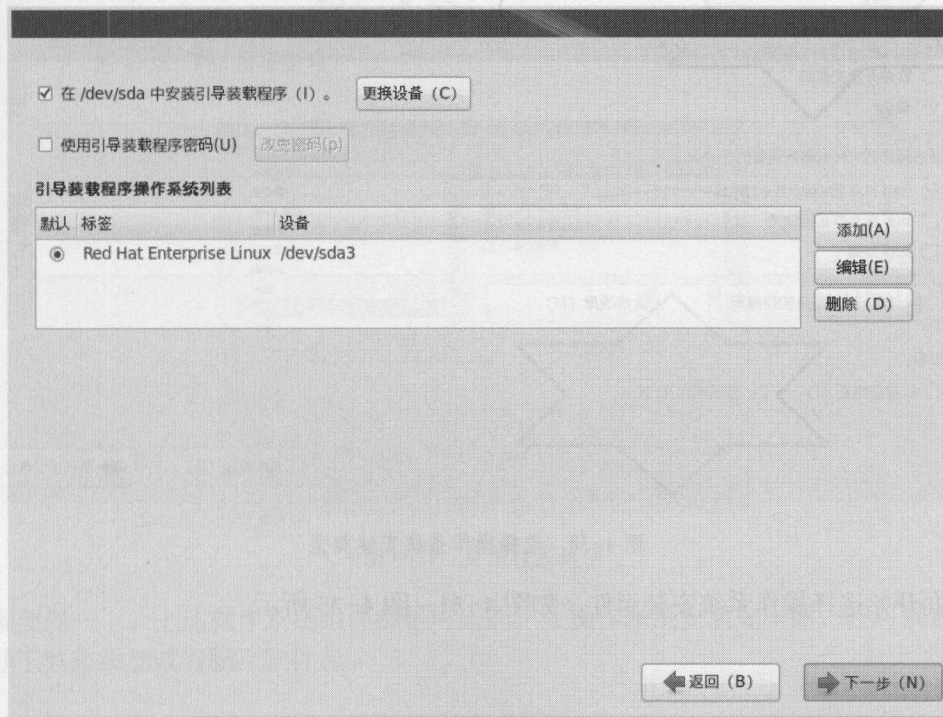


图 4-29 装载引导程序

4.8.2 选择操作系统安装类型

可选的类型说明如下：

- 基本服务器：安装的基本系统的平台支持，不包含桌面。
- 数据库服务器：基本系统平台，加上 MySQL 和 PostgreSQL 数据库，无桌面。
- 万维网服务器：基本系统平台，加上 PHP，Web server，还有 MySQL 和 PostgreSQL 数据库的客户端，无桌面。
- 虚拟化主机：基本系统加虚拟平台。
- 桌面：基本的桌面系统，包括常用的桌面软件，如文档查看工具。
- 软件开发工作站：包含软件包较多，基本系统，虚拟化平台，桌面环境，开发工具。
- 最小：最基本的操作系统，不包含任何可选的软件包。

在此我们选择“桌面”类型，以及“现在自定义”选项，并手工选择需要安装的操作系統组件，如图 4-30 所示。

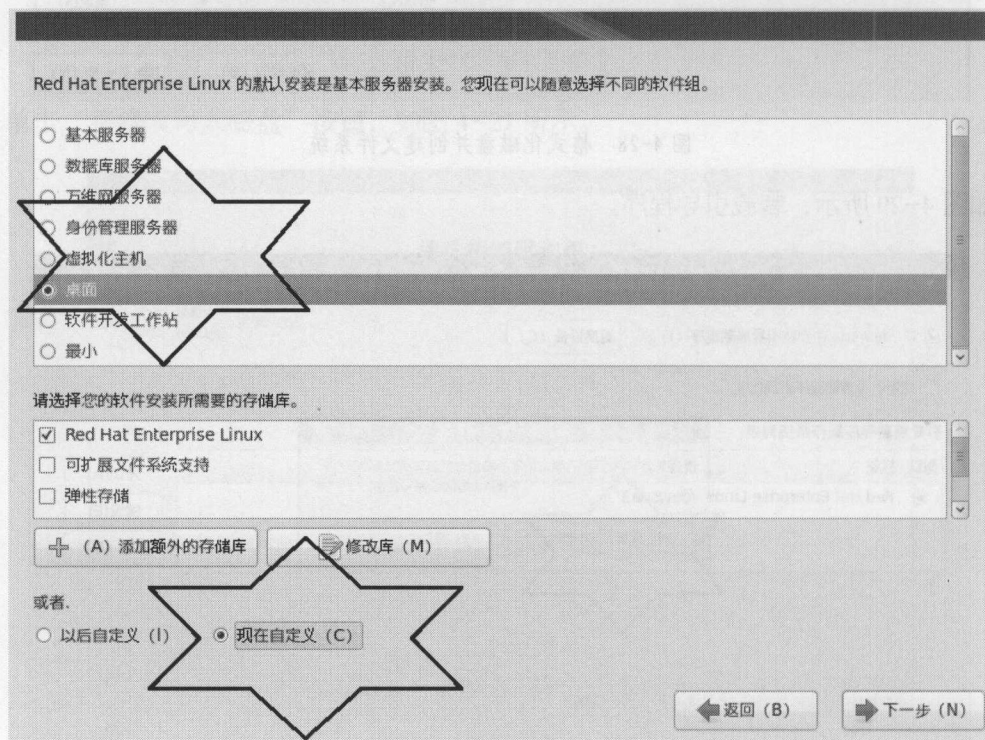


图 4-30 选择操作系统安装类型

下面开始选择操作系统安装组件，如图 4-31～图 4-35 所示。

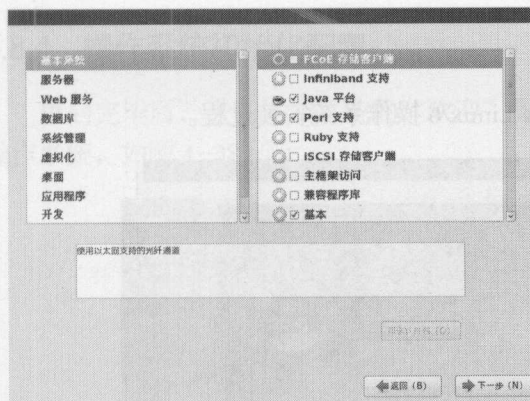


图 4-31 基本系统选项

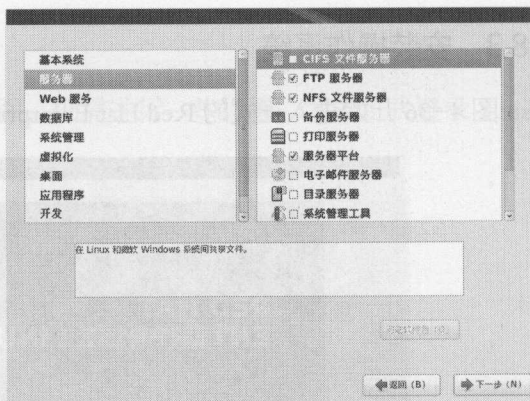


图 4-32 服务器选项

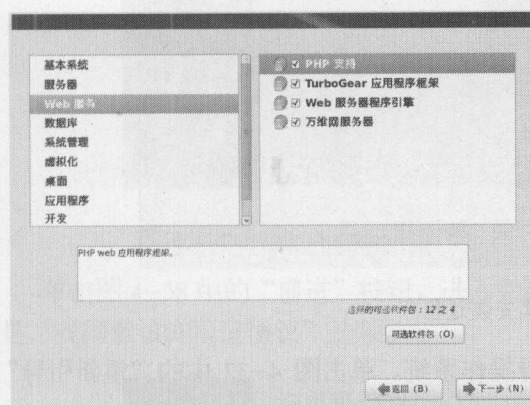


图 4-33 Web 服务选项

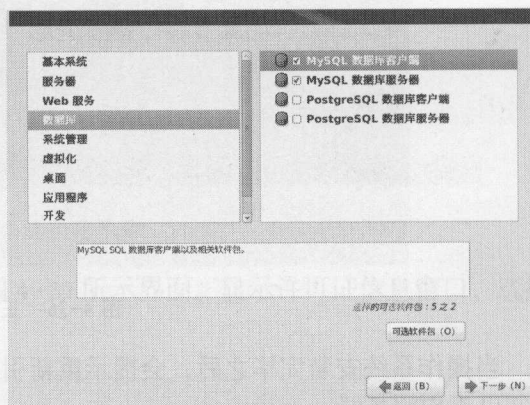


图 4-34 数据库选项

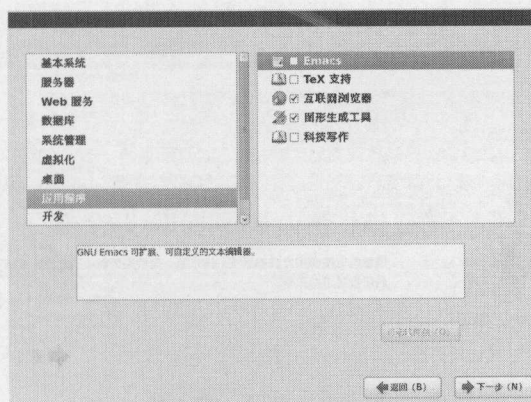


图 4-35 应用程序选项

顺便提一下，上述操作系统组件都是可选的。在操作系统安装完毕后，还可以通过软件包管理工具来添加或者删除软件包。

4.8.3 安装操作系统

图 4-36 开始进入正式的 Red Hat Enterprise Linux 6 操作系统安装过程。

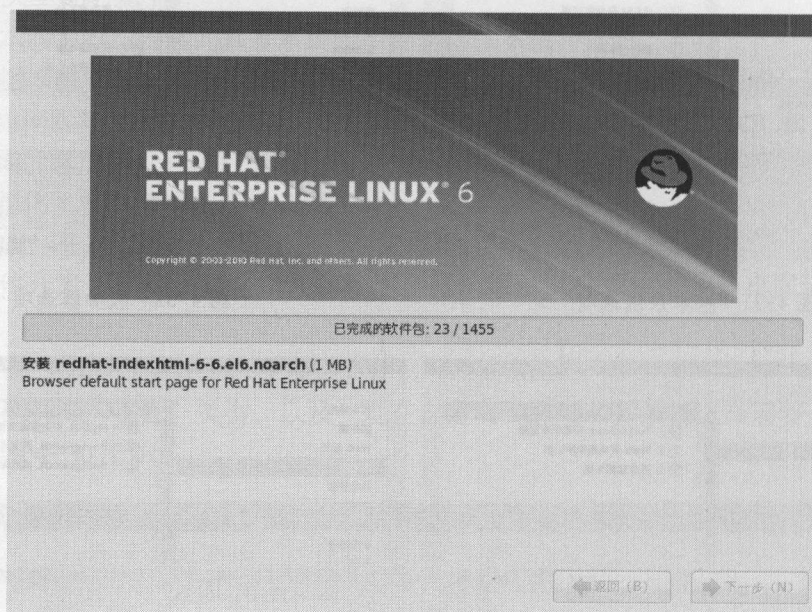


图 4-36 正式安装过程

当操作系统安装完毕之后，会提示重新引导操作系统。单击图 4-37 中的“重新引导”按钮以重启服务器。

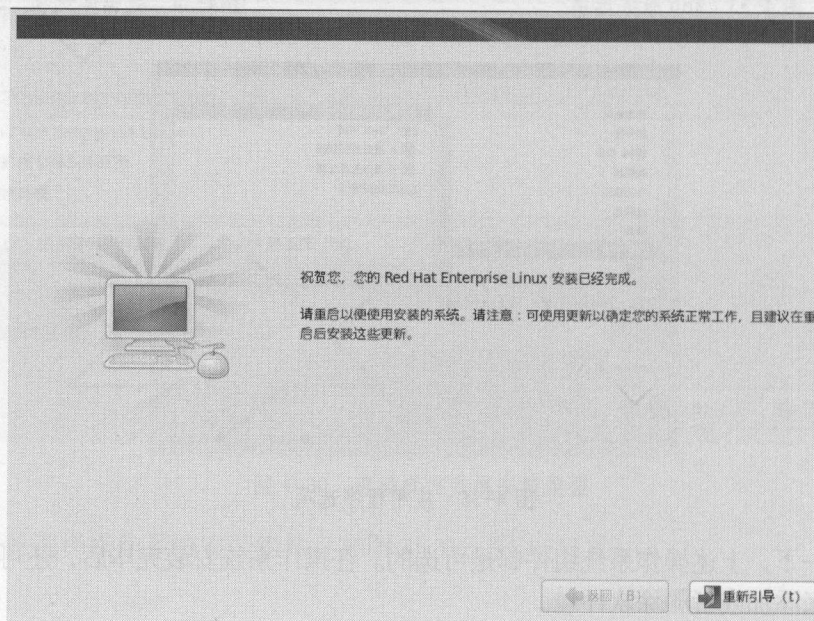


图 4-37 重新引导操作系统

4.8.4 操作系统初始化配置

重启完毕后，进入操作系统的“欢迎”界面，仍需进行一些设置，才能正式登录 Linux 操作系统，如图 4-38 所示。



图 4-38 操作系统欢迎界面

单击图 4-38 中的“前进”按钮，进入如图 4-39 所示界面。显示许可证信息窗口，选择“是，我同意该许可证协议”选项。

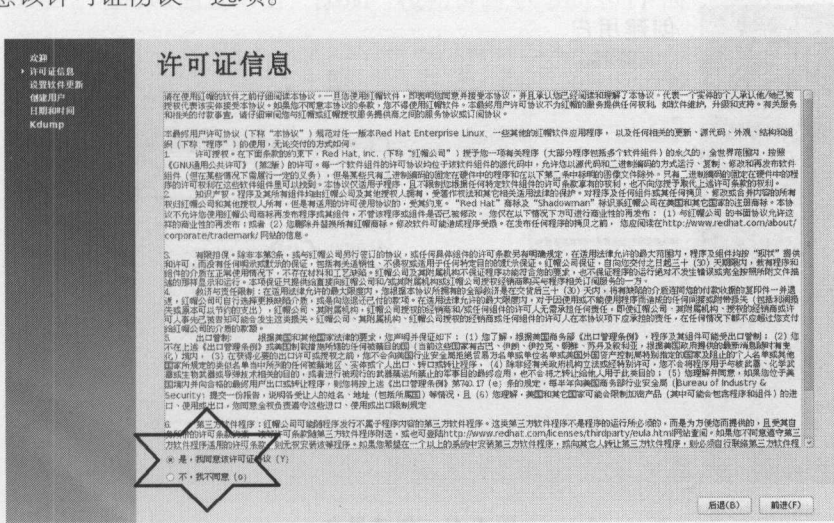


图 4-39 同意许可协议

单击图 4-39 中的“前进”按钮，进入图 4-40 中的“设置软件更新”页面。在该页面中，可以注册您的红帽账号，以便于登录红帽网络，进行系统软件更新等操作。由于我们安装的是服务器系统，运行 Nagios 等开源监控软件，如果未经证实或者测试，一般不需要更新操作系统软件，以免发生兼容性方面的问题。因此我们选择“不，以后再注册”选项。

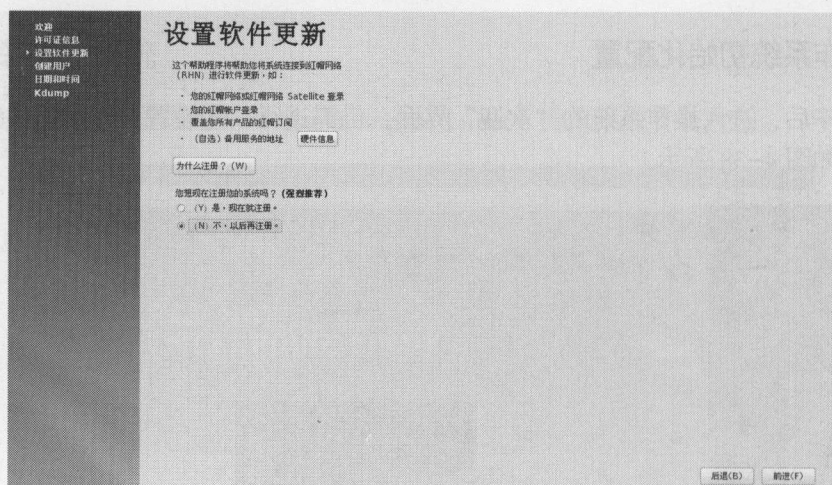


图 4-40 设置软件更新

4.8.5 创建操作系统账户

单击图 4-40 中的“前进”按钮，进入如图 4-41 所示界面，在这里用户可以通过输入用户名、全称和口令创建一个普通用户的账号。在此，我们创建了“monitor”用户，并且在“高级”选项卡里设置了 monitor 用户的主目录为/home/monitor。在“用户管理者”窗口中，您可以创建多个组、用户。创建完毕后，直接单击“前进”按钮。

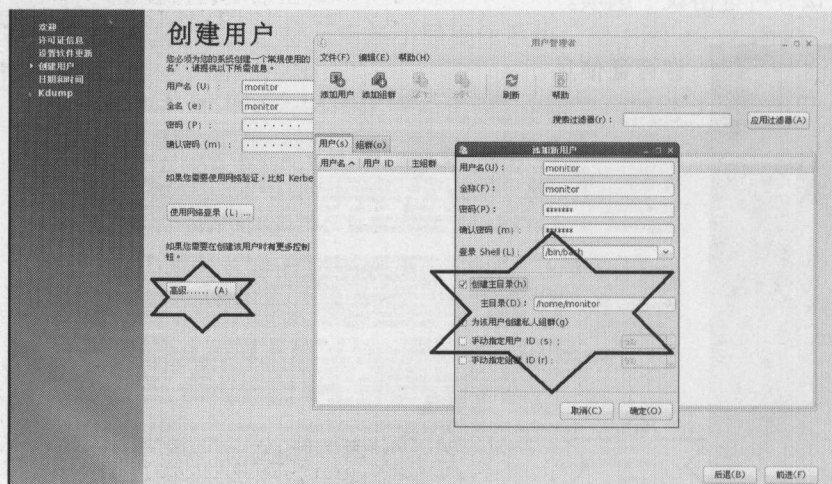


图 4-41 创建用户

4.8.6 设置操作系统时间

在如图 4-42 所示界面上，用户可以手工配置计算机系统的日期和时间，也可以通过连接在互联网上的网络时间服务器（NTP 服务器）为本机传输日期和时间信息，并且可以和 NTP 服务器的时间同步。要启用时间同步的功能，选择“在网络上同步日期和时间”选项即可，配置完毕单击“前进”按钮。

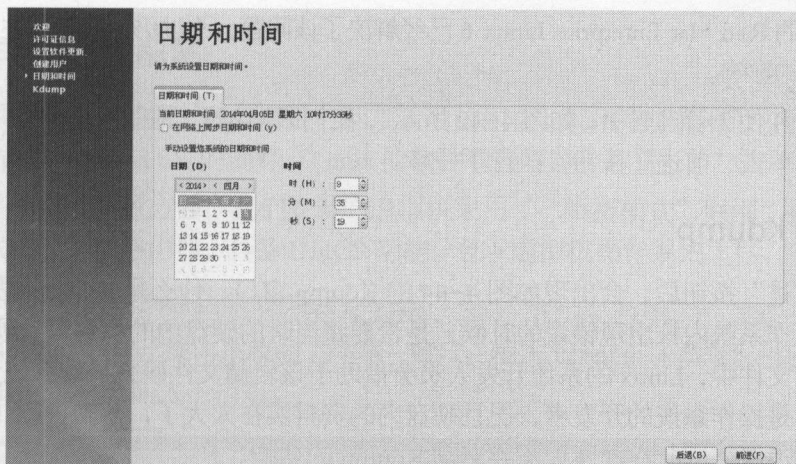


图 4-42 设置日期与时间

如图 4-42 所示，如果选择“在网络上同步日期和时间”选项，将会开启 Linux 操作系统的 NTP 服务。NTP 服务“Network Time Protocol (NTP)”是用来使计算机时间同步化的一种协议，操作系统借助于 NTP 服务，可以从多个时间来源，例如原子钟、天文台、卫星、位于 Internet 或者 Intranet 上的时钟服务器（即提供授时服务的 NTP 服务器）上获取国际标准时间（UTC）信息。

如图 4-43，选择“在网络上同步日期和时间”选项之后，操作系统会提供多个 Internet 上的 NTP 服务器，还可以通过单击“添加”按钮设置更多的 NTP 时钟服务器。

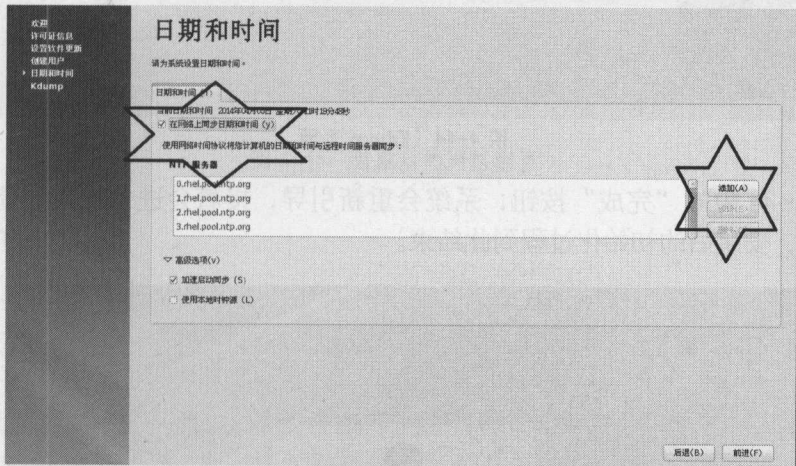


图 4-43 NTP 服务器

值得注意的是，一旦为 Linux 操作系统开通了 NTP 服务，且与 NTP 服务器保持时间同步，您就需要特别关注安全问题，例如著名的“闰秒”问题（请参照维基百科 <http://zh.wikipedia.org/wiki/闰秒>）。闰秒问题将影响部分开启 NTP 服务的 Linux 操作系统——会导致 Linux 内核崩溃！Linux kernel 是在 2.6.18-164.el5 之后的版本中解决了这个问题。换句话说，Linux kernel 低于 2.6.18-164 的 Linux 系统，无论是什么公司的 Linux 都将受到影响。当然，

本次我们安装的 Red Hat Enterprise Linux 6 已经解决了该问题，不过仍旧需要关注 NTP 服务对我们可能造成的影响。

为了安全和便于管理起见，如图 4-42 所示，我们取消对于“在网络上同步日期和时间”选项的选择，单击“前进”按钮，执行下一步。

4.8.7 设置 Kdump

单击“前进”按钮后，会出现如图 4-44 的 Kdump 窗口。什么是 Kdump 呢？Kdump 就是当 Linux 操作系统内核出现错误的时候，是否要将当时的硬盘内的信息写到硬盘上的操作系统内核转储文件中，Linux 的系统开发人员会借助于该转储文件研究操作系统为何会崩溃。由于我们并不是操作系统的开发者，况且硬盘内的资料实在太大了，常常进行 Kdump 会造成硬盘空间的浪费，因此在此建议不要启动 Kdump 的功能。

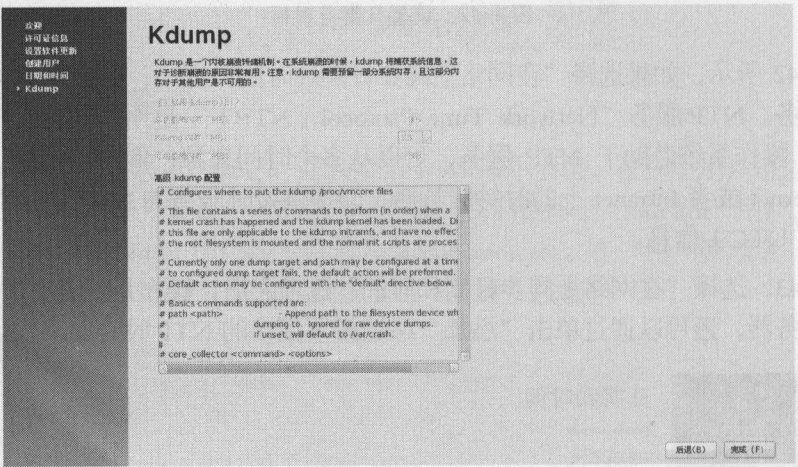


图 4-44 Kdump 页面

单击图 4-44 中的“完成”按钮，系统会重新引导，接着会进入正式的登录界面。如图 4-45 所示，安装后的初始化过程到此结束。

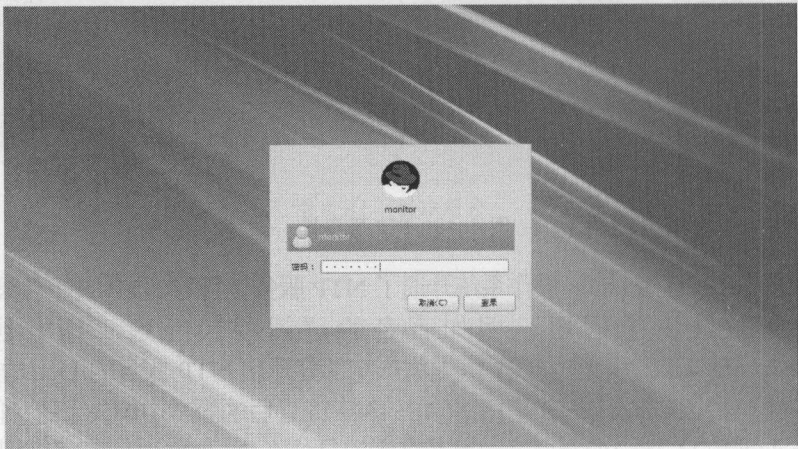



图 4-45 登录界面

4.8.8 操作系统网络配置

当操作系统安装完毕后,还需要进行两项重要的配置,即网络配置和软件源(即 yum, Yellow dog Updater, Modifier 的简称, Linux 的软件包管理器)配置。

在操作系统网络配置方面,我们安装的虚拟机采用了“桥接模式”联网,且虚拟机所在的物理机上联的是一台路由器,具备 DHCP 功能。登录虚拟机操作系统之后,单击右上角的  图标,即可自动从路由器获取 IP 地址。

如图 4-46 所示,打开操作系统的“终端”窗口,输入 `ifconfig -a` 命令,可以看到系统已经获取到了 IP 地址,无需人工指定。

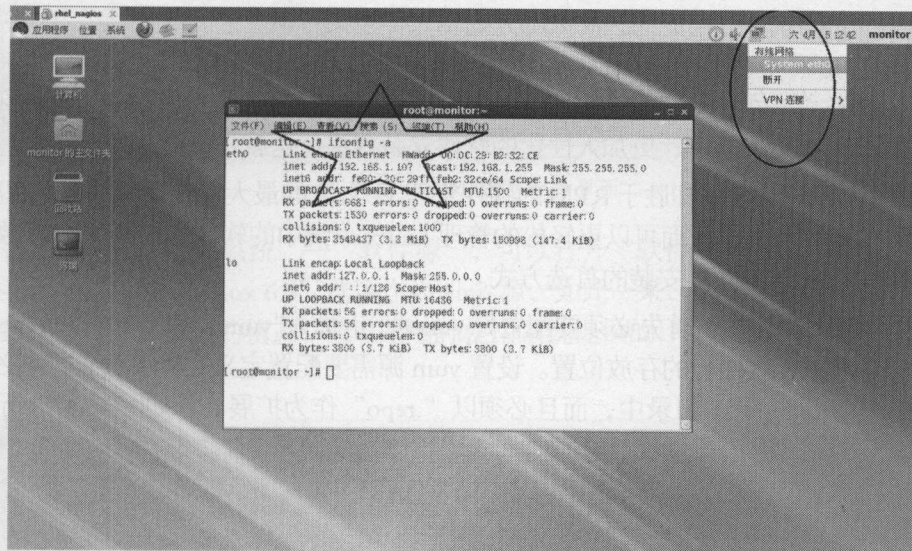


图 4-46 登录后的网络配置

4.8.9 yum 源配置

说起 yum 源,首先要了解什么是 yum。yum 是一款在 Linux 操作系统中比较流行的软件包管理器,它能帮助管理员更方便地添加、删除和更新 RPM 包。如果我们只是下载了单个 RPM 包,由于 RPM 包之间存在依赖关系,总是提醒我们要下载并安装其他的 RPM 包。而 yum 能够帮助人们从指定的服务器自动下载 RPM 包并且安装,可以自动处理依赖性关系,并且一次安装所有依赖的软件包,无需繁琐地一次次下载、安装。yum 提供了查找、安装、删除某一个、一组甚至全部软件包的命令,而且命令简洁而又好记。

在没有以 yum 方式安装 rpm 软件包之前, Linux 上还存在源代码安装和 RPM 包安装两种方式。源代码安装方式是最原始的安装方式,这种方法虽然古老并且复杂,但仍然有很多人在用。这是由于在 Linux 系统中使用的绝大多数软件都是开源软件,软件作者在发布软件时直接提供的就是软件的源代码。用户在取得应用程序的源码文件后,可以根据自身需求对软件进行修改或定制,然后在自己的系统上重新编译,即可生成能在该系统上执行的程序文件。通过源码安装,用户可以获得最新的应用程序,可以定制更灵活、丰富的功能,而且使

软件可以跨越计算机平台，在所有版本的 Linux 系统中都能使用。例如，在第 2 章开篇就提到，我们要从源代码开始编译并安装 Nagios 软件，从而获得更大的灵活性和平台一致性。

对于大多数用户而言，源码安装方式过于复杂，耗时又长，对用户的软件开发能力要求也比较高。为此 Red Hat 特别设计了一种名为 RPM (Red Hat Packet Manager) 的软件包管理系统，RPM 是一种已经编译并封装好的软件包，用户可以直接安装使用。通过 RPM，用户可以更加轻松方便地管理系统中的所有软件。RPM 软件包只能在使用 RPM 机制的 Linux 操作系统中使用，如 RHEL、Fedora、Suse 等。在 Linux 世界中，还有另外一种名为 DEB 的软件包管理机制，可以在 Debian、Ubuntu 等系统中使用。相比较而言，还是 RPM 安装包应用更为广泛，基本已成为 Linux 系统中软件安装包事实上的标准。但是 RPM 也有一个很大的缺点，即 RPM 软件包之间存在着复杂的依赖关系。在多数情况下，一个软件都是由多个相互依赖的 RPM 软件包组成的，而大部分的 RPM 包又有相互之间的依赖关系。例如，安装 A 软件需要 B 软件的支持，而安装 B 软件又需要 C 软件的支持，那么在安装 A 软件之前，必须先安装 C 软件，再安装 B 软件，最后才能安装 A，有时甚至还可能会出现死循环。所以后来又出现了一种更加简单、更加人性化的软件安装方法，这就是 yum 安装。

yum 是一个基于 RPM 却胜于 RPM 的软件管理工具，它的最大优点是可以自动解决 RPM 软件包间的依赖性问题，从而可以更轻松的管理 Linux 系统中的软件。从 RHEL5 开始，Red Hat 就推荐用 yum 作为软件安装的首选方式。

要使用 yum 安装方式，首先必须要配置好 yum 源(也称为“yum 仓库”，即 yum repository)。yum 源是所有 RPM 软件包的存放位置。设置 yum 源需要配置定义文件，定义文件必须存放在指定的 /etc/yum.repos.d/ 目录中，而且必须以 “.repo” 作为扩展名，如图 4-47 所示。



图 4-47 yum 配置文件

一般来说，对于普通用户，我们使用操作系统安装光盘作为 yum 源就可以了；不需要另外指定 yum 源。尤其是服务器运行环境，不同于桌面环境，安装不同来源的软件容易引发兼容性问题。

我们只要将虚拟机光盘挂载到文件系统上，以 root 用户登陆操作系统，选择“系统”→“管理”→“添加和删除软件”，可以进入红帽的“添加/删除软件”界面。

图 4-48 所示的界面是红帽 Linux 的 PackageKit 应用窗口，是采用 yum 源安装方式的图形界面。PackageKit 能够指向多个 yum 源，并且获取 yum 源上能够安装的 RPM 软件包。

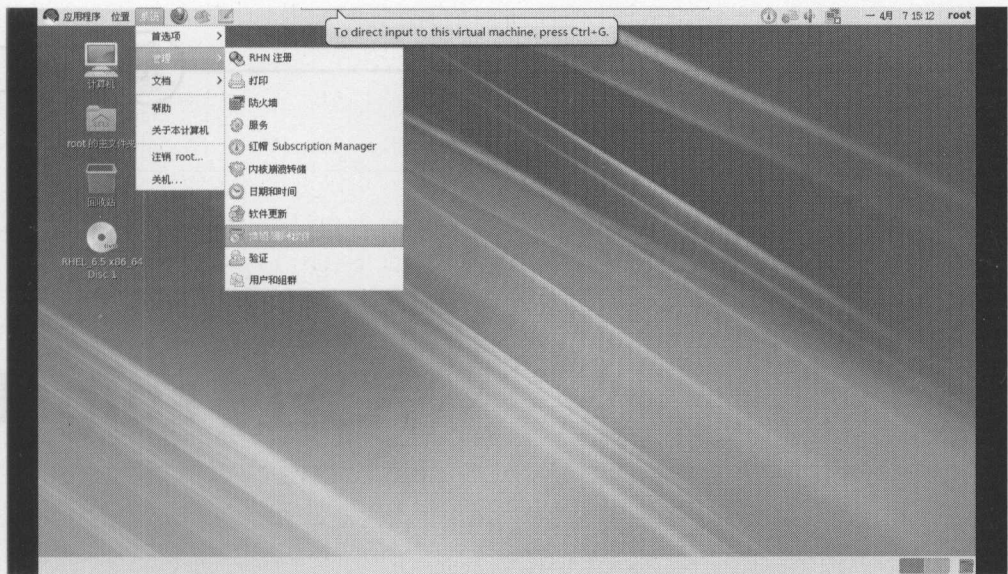


图 4-48 添加/删除软件界面

单击图 4-49 中的“系统”→“软件源”，可以打开“软件源”窗口。在这里，可以选择“Red Hat Enterprise Linux 6.5”选项作为 yum 源，如此一来，只要一直保持操作系统光驱为“挂载”且随时可访问的状态，就可以随时使用系统提供的“添加/删除软件”工具来安装需要的 RPM 软件包了。

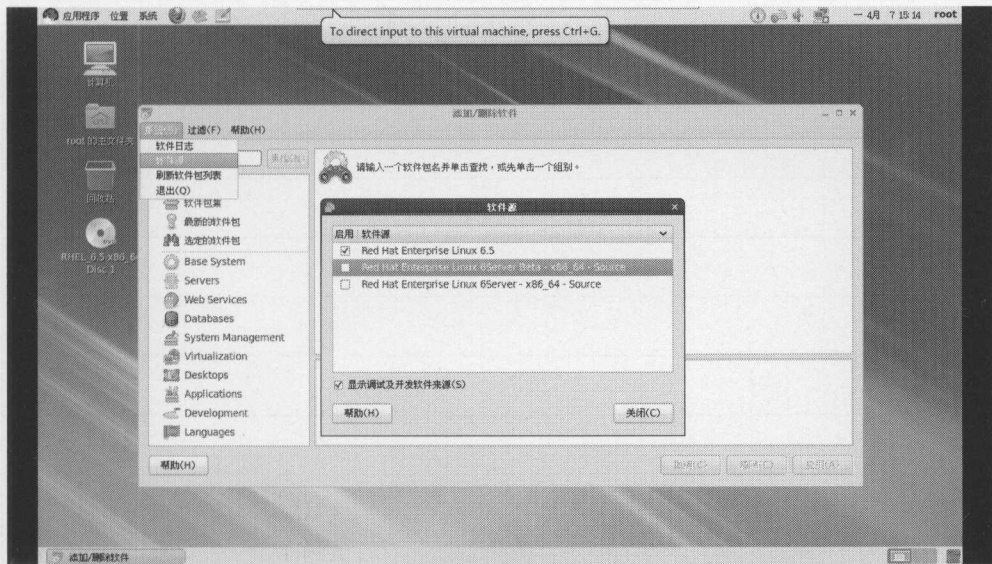


图 4-49 显示并选择安装光盘为 yum 源



读书笔记

Large rounded rectangular area with horizontal dashed lines for taking notes.

第5章

Nagios 的安装

在开始一次外出旅行之前，我们需要做充足的准备，尽量避免出行后的手忙脚乱、丢三落四，同样地，安装 Nagios 之前，我们也要规划好大致的安装步骤，防止安装后的零碎问题造成我们的顾此失彼。做好准备工作，不敢说完美无缺，但至少能够掌握主动权。

5.1 Nagios 安装前的准备工作

正如第 1 章开篇所述，“安装 Nagios 的最简单方式就是借助于 Linux 发行版中的软件包安装工具”。从 Nagios 的 2.0 版本开始，就已经成熟到足以成为 Linux 发行版的重要组件了。正因为 Linux 发行版的差别，导致操作系统中 Nagios 软件包的默认安装路径各有不同，既造成了与 Nagios 源代码中的原始安装路径不一致，也为后期管理带来不便。

正因如此，当遇到版本较新的 Nagios 3.0 甚至 Nagios 4.0 以上版本时，还是建议大家不要使用操作系统提供的 Nagios 安装包来安装和部署 Nagios（如果已经安装了，建议通过操作系统提供的软件管理工具进行卸载）。最好从源代码开始，编译并部署一套全新的 Nagios 系统，从中获取新鲜感和成就感。换句话说，如果从源代码开始编译和部署 Nagios，用户对于 Nagios 的目录结构、各组件的位置和参数配置会有一个全面的掌握。另一方面，采用这种方式编译出来的 Nagios，各项参数配置信息都是完整且未被修改过的，用户将会得到一个“干净”的 Nagios 运行环境。

当然了，从源代码编译 Nagios 不是一项轻松的工作，尤其是在 Linux 操作系统上，需要在系统里安装一些必须的软件运行环境和可选的软件包。首先需要注意的是各软件的版本号，表 5-1 是本次安装过程选用的软件及版本号。

表 5-1 安装所需软件及版本号

所需软件	版本号
Linux 操作系统	Red Hat Enterprise Linux 6.5 X86_64
操作系统软件包	mysql、httpd、gcc、glibc、glibc-common、gd、gd-devel
Nagios	3.4.3
Nagios 插件(Nagios plugins)	1.4.12
nrpe	2.12
ndoutils	1.5
mysql	5.6
Centreon	2.4.3
NagVis	1.7.3

为了编译和运行 Nagios，您需要检查、提前下载并安装一些必要的编译工具和运行库，例如 gcc、make、autoconf、automake 工具，libgd 和 openssl 等运行库，这些运行库的开发（development）包也必须安装（运行库的开发包通常以 -dev、-devel 结尾）：如 libssl-dev、libgd-dev 和 libc6-dev 软件包。

对于红帽操作系统，我们按照图 4-48 所述进入操作系统的“添加/删除软件”界面，在左上角的“查找”栏输入编译工具和库文件关键字，并单击“查找”按钮，即可列出所有包含关键字的软件包，选择并安装即可。一般来说，我们需要查找并安装包含以下关键字的所有软件包：httpd、gcc、glibc、glibc-common、gd、gd-devel，如图 5-1 所示。

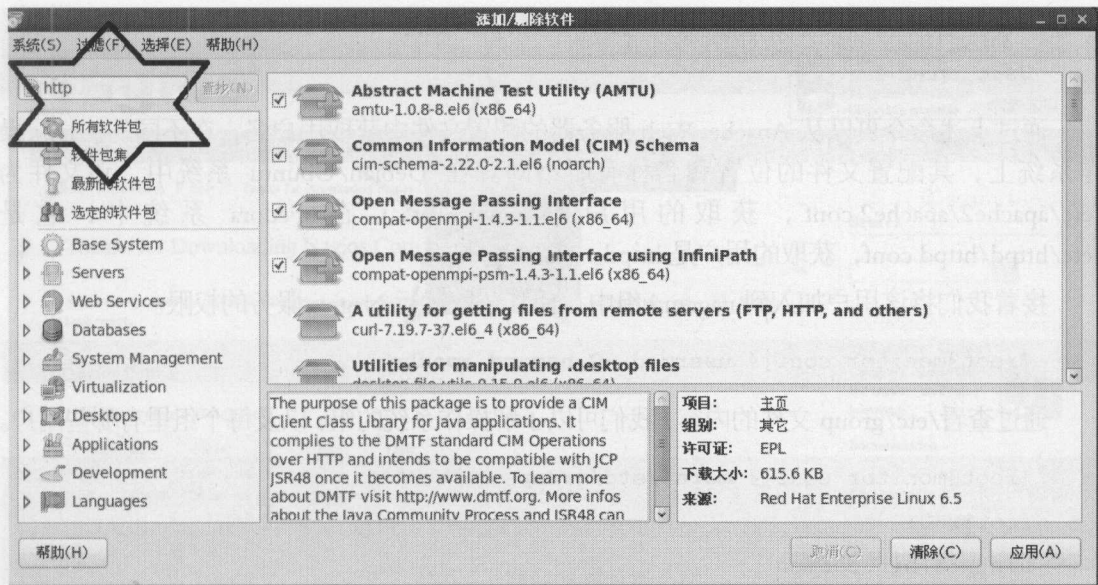


图 5-1 安装库文件

5.2 创建 Nagios 用户和组

在正式编译和安装 Nagios 之前,需要在操作系统上创建 Nagios 用户和组,我们指定 Nagios 用户名为 nagios, 主要属组为 nagios, 同时属于 nagcmd 组。

使用命令 `groupadd` 来创建组 nagios 和 nagcmd, 使用命令 `useradd` 来创建用户 nagios。为了使 nagios 用户有权限运行 Nagios 服务, 必须将 nagios 用户加入到 nagios 组和 nagcmd 组里。

```
[root@monitor ~]# groupadd -g 9000 nagios
[root@monitor ~]# groupadd -g 9001 nagcmd
[root@monitor ~]# useradd -u 9000 -g nagios -G nagcmd -d /home/nagios -c
"Nagios Admin" nagios
```

在上述命令中, 我们创建了 ID 为 9000 的 nagios 组以及 ID 为 9001 的 nagcmd 组, 然后创建了 ID 为 9000 的 nagios 用户, 将 nagios 用户的起始群组 (initial group) 设置为 nagios, 并将其同时加入 nagcmd 组。在此我们选用的 ID 值也可以设置为其他未曾用过的 ID 值。由于 nagios 组具备管理 Nagios 服务的重要权限, 因此配置为仅拥有 nagios 这一个用户, 不会再加入其他用户。

Nagios 的重要组成部分是位于服务器端的一系列 CGI 程序, 这些 CGI 程序是由 Apache Web 服务器来运行的, 因此 Apache Web 服务器的用户也应该有权限来访问和执行 Nagios 用户组下的 CGI 程序。为了确保 Nagios 服务器端 CGI 程序的安全, 我们专门创建了 nagcmd 组, 只有 Nagios 用户和 Apache Web 服务器的用户属于该组。前面我们在创建 nagios 用户的同时, 就已经将其加入到 nagcmd 组中, 下面我们先确定 Apache Web 服务器的用户, 再将其加入 nagcmd 中。

使用如下命令查看 Linux 操作系统上的 Apache Web 服务器用户。

```
[root@monitor conf]# grep "^User" httpd.conf
User apache
```

通过上述命令可以从 Apache Web 服务器的配置文件中获取用户名，在不同的 Linux 操作系统上，其配置文件的位置和名称有所不同。在 Debian/Ubuntu 系统中，该文件为 /etc/apache2/apache2.conf，获取的用户是 www-data；在 Fedora 系统中，它是 /etc/httpd/httpd.conf，获取的用户是 httpd。

接着我们将该用户加入到 nagcmd 组中，使其获取运行 Nagios 服务的权限。

```
[root@monitor conf]# usermod -G nagcmd apache
```

通过查看 /etc/group 文件的内容，我们可以了解操作系统的组，以及每个组里有哪些用户。

```
[root@monitor conf]# more /etc/group
root:x:0:
bin:x:1:bin,daemon
daemon:x:2:bin,daemon
sys:x:3:bin,adm
adm:x:4:adm,daemon
tty:x:5:
disk:x:6:
```

在 /etc/group 文件的末尾，我们可以看到新加入的 nagios、nagcmd 组以及 nagios 用户。

```
nagios:x:9000:
nagcmd:x:9001:nagios,apache
```

在创建 nagios 用户时，我们顺便创建了 nagios 用户的主目录为 /home/nagios，接下来我们要创建 Nagios 服务的安装目录，指定为 /usr/local/nagios，并赋予权限。

```
[root@monitor nagios]# mkdir /usr/local/nagios
[root@monitor nagios]# chown -R nagios:nagios /usr/local/nagios
```

在某些版本的 Linux 操作系统上，为了遵循操作系统对于目录结构的组织以及管理，可以将 Nagios 的配置文件放在 /etc/nagios 目录下，这就意味着除了 /usr/local/nagios 目录外，用户还需要另外创建 /etc/nagios 和 /var/nagios 目录并赋予读写和属主权限。在本书中，我们将 Nagios 的程序文件、配置文件和运行数据文件统一放到 /usr/local/nagios 目录下，便于统一管理。

5.3 编译并安装 Nagios

在 Nagios 的官方网站（www.nagios.org）里，可以找到所有关于 Nagios 的信息，包括 Nagios 的源代码。Nagios 相关的源代码可以在网站的“Projects”栏里找到，如图 5-2 所示。

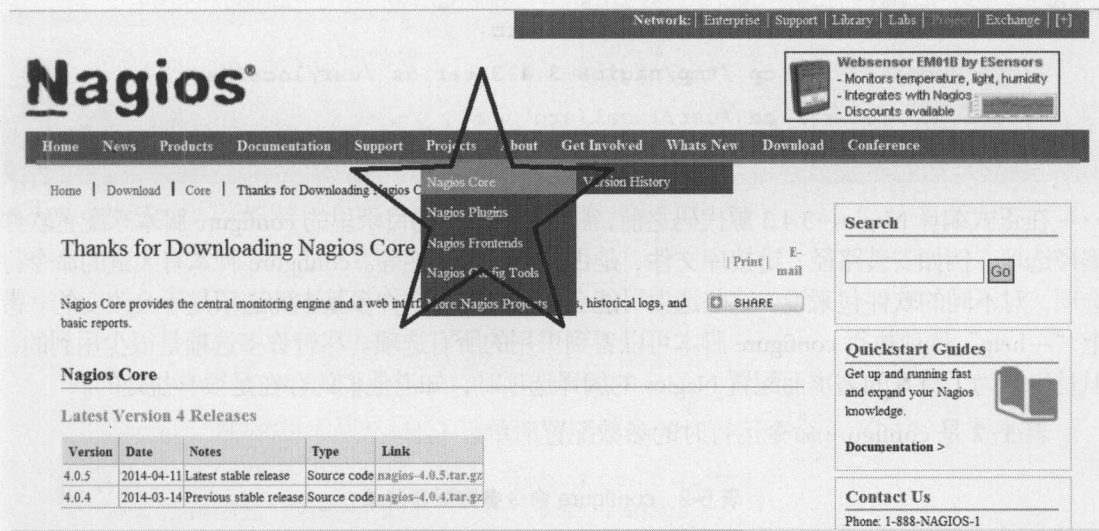


图 5-2 Nagios 源代码下载

各项详述如下：

- **Nagios Core**：Nagios 核心引擎，提供监控项调度、监控事件处理、告警信息管理等 Nagios 核心功能。出于性能考虑，Nagios Core 由 C 语言编写，以后台进程（daemon）运行在 Linux 以及其他类 Unix 操作系统上。
- **Nagios Plugins**：Nagios 的监控插件，是 Nagios 核心引擎的扩展。Nagios 核心引擎自身并不负责执行具体的监控功能，而是借助各色插件来实现不同平台上的监控。Nagios 插件一般位于被监控节点的特定目录下，是一系列独立的小程序，自身可以接收参数并按照 Nagios 引擎可以识别的格式来输出监控结果。Nagios 插件可以是编译好的可执行文件，也可以是由 shell、Perl、PHP 或者 Python 写成的脚本。
- **Nagios Frontends**：即 Nagios 前端界面。由于 Nagios 官方长期专注于核心引擎部分的发展，而忽略了前端界面的更新。故许多 Nagios 相关的开源项目都将注意力放在了前端界面展示的开发工作上，旨在提供具有现代感的 Nagios 前端用户界面，甚至移动 APP 界面，以替代 Nagios 落后的 CGI 网页界面。
- **Nagios Config Tools**：Nagios 配置工具。若要 Nagios 能够正常工作，需要对 Nagios 核心引擎做各种个性化配置。目前 Nagios 核心引擎的配置都是通过启动时读入文本文件执行的，采用文本文件配置 Nagios 核心引擎的方式，在为专业的系统管理员提供了最大限度灵活性的同时，也给经验不足的普通用户管理 Nagios 带来了诸多不便，稍不留神就会造成配置文件条目的不匹配，导致 Nagios 核心引擎无法正常启动。对于普通的 Nagios 用户而言，Nagios 配置工具提供了现代化的 Nagios 配置及管理界面，包括监控主机配置、监控项配置、用户配置、监控命令配置、模板配置等管理复杂配置项，以及装载和写入 Nagios 配置文件的功能。

在本书中，根据表 5-1 的规划，我们安装的 Nagios 的 3.4.3 版本，旧的 Nagios 2.0 版本以及新的 4.0 以上版本安装方式和步骤差别不大。

接下来的步骤中，我们将下载到的 nagios-3.4.3.tar.gz 源代码文件放在/tmp 目录下，创建 /usr/local/src 目录，将 nagios-3.4.3.tar.gz 文件解压至该目录，然后开始编译并安装 Nagios。


```
[root@monitor /]# mkdir /usr/local/src
[root@monitor /]# cp /tmp/nagios-3.4.3.tar.gz /usr/local/src
[root@monitor /]# cd /usr/local/src
[root@monitor src]# tar xvzf nagios-3.4.3.tar.gz
```

在正式编译 Nagios-3.4.3 源代码之前，需要运行源代码目录里的 `configure` 脚本来配置软件编译选项，例如安装路径、链接库文件、是否开启某项功能等。`configure` 脚本有大量的命令行选项，对不同的软件包来说，这些选项可能会有变化，但是许多基本的选项是不会改变的。带上“`-help`”选项执行 `configure` 脚本可以看到可用的所有选项。尽管许多选项是很少用到的，但是当你为了特殊的需求而配置 Nagios 的编译选项时，知道他们的存在是很有益处的。

表 5-2 是 `configure` 命令运行时的参数配置清单。

表 5-2 `configure` 命令参数配置清单

属 性 名	参 数 值	configure 参数
Nagios 安装目录	/usr/local/nagios	--prefix
Nagios 配置文件目录	/usr/local/nagios/etc	--sysconfdir
Nagios 运行数据目录	/usr/local/nagios/var	--localstatedir
Nagios 用户	nagios (9000)	--with-nagios-user
Nagios 用户组	nagios(9000)	--with-nagios-group
Nagios 命令组	nagcmd(9001)	--with-command-group
开启 Nagios 事件代理的 Perl 解释器（可以在 Nagios 运行时，使事件代理 Event Broker 模块提供对外调用接口）		--enable-embedded-perl
开启 Perl 缓存		--with-perlcache

如上表所述，`--prefix` 参数决定了 Nagios 的安装路径，此项不建议修改，保持为默认的 `/usr/local/nagios` 目录。

而 Nagios 配置文件目录和 Nagios 运行数据（通常是一些 Nagios 运行过程中产生的 `log` 日志文件、`status` 状态文件等）目录分别由 `--sysconfdir` 和 `--localstatedir` 决定。Nagios 默认分别是 `/usr/local/nagios/etc` 和 `/usr/local/nagios/var` 目录。需要指出的是，根据 2.1 节所述的 FHS 标准，Linux 操作系统应用程序的配置目录和运行时产生的数据目录应该分别是根目录下的 `/etc` 和 `/var`，而在本书中为管理方便起见，这两项配置不必人工指定，仍采取 Nagios 默认的目录配置。

综上所述，我们进入 Nagios 的源代码目录，执行 `configure` 命令。

```
[root@monitor /]# cd /usr/local/src/nagios
[root@monitor nagios]# ./configure --prefix=/usr/local/nagios
--with-nagios-user=nagios --with-nagios-group=nagios
```

```
--with-command-group=nagcmd --enable-embedded-perl --with-perlcache
```

以下是 configure 命令执行结果。

```
*** Configuration summary for nagios 3.4.3 11-30-2012 ***:
```

```
General Options:
```

```
-----
```

```
Nagios executable: nagios
```

```
Nagios user/group: nagios,nagios
```

```
Command user/group: nagios,nagcmd
```

```
Embedded Perl: yes, with caching
```

```
Event Broker: yes
```

```
Install ${prefix}: /usr/local/nagios
```

```
Lock file: ${prefix}/var/nagios.lock
```

```
Check result directory: ${prefix}/var/spool/checkresults
```

```
Init directory: /etc/rc.d/init.d
```

```
Apache conf.d directory: /etc/httpd/conf.d
```

```
Mail program: /bin/mail
```

```
Host OS: linux-gnu
```

```
Web Interface Options:
```

```
-----
```

```
HTML URL: http://localhost/nagios/
```

```
CGI URL: http://localhost/nagios/cgi-bin/
```

```
Traceroute (used by WAP): /bin/traceroute
```

```
Review the options above for accuracy. If they look okay,
type 'make all' to compile the main program and CGIs.
```

检视上述的 configure 配置结果清单，如果不出问题的话，接着依次执行下述 6 个命令，编译并安装 Nagios。

```
[root@monitor nagios]# make all
```

```
[root@monitor nagios]# make install
```

```
[root@monitor nagios]# make install-init
```

```
[root@monitor nagios]# make install-commandmode
```

```
[root@monitor nagios]# make install-webconf
```

```
[root@monitor nagios]# make install-config
```

上述命令编译所有 Nagios 相关的程序组件，包括 CGI 程序、文档等，并将其安装到之

前所配置的目录中。值得注意的是 `make install-init` 命令，它会将 Nagios 的启动、停止相关的脚本文件安装到系统的 `/etc/init.d` 目录中，这样通过操作系统的 `service` 命令可以管理 Nagios 进程的启动、停止工作。

安装完毕后，Nagios 运行相关的目录都在 `/usr/local/nagios` 下，清单如表 5-3 所示。

表 5-3 在 `/usr/local/nagios` 下的目录清单

目 录	内容清单
<code>./bin</code>	Nagios 可执行程序所在目录
<code>./libexec</code>	外部插件存放目录
<code>./sbin</code>	CGI 脚本
<code>./share</code>	文档，HTML 文件存放目录
<code>./etc</code>	配置文件目录
<code>./var</code>	运行时日志、数据文件、lock 文件所在目录
<code>./var/archives</code>	Nagios 日志自动归档目录
<code>./var/rw</code>	用来存放外部命令文件的目录

5.4 安装 Nagios 插件

接下来我们开始从源代码开始编译并安装 Nagios 插件。如图 5-2所示，Nagios 插件的最新版本仍旧可以从 Nagios 的官方网站（www.nagios.org）里下载，目前版本是 2.0，其历史版本可以从 Nagos-Plugin 的网站（<http://nagios-plugins.org/download/>）里下载。本书使用的版本是 `nagios-plugins-1.4.16` 版本。Nagios 插件是与 Nagios 分开的独立组件，这就意味着新版本的 Nagios 可以兼容旧版本的 Nagios 插件。另外，新旧版本的 Nagios 插件也具备相互兼容性，如果你对新版本的 Nagios 插件中的某项插件感到不满意，完全可以用旧版本中相对应的项来替代。

接下来，我们将下载的 `nagios-plugins-1.4.16.tar.gz` 放到 `/tmp` 中，然后解压到 `/usr/local/src` 目录下配置并安装。

```
[root@monitor tmp]#cp /tmp/nagios-plugins-1.4.16.tar.gz /usr/local/src
[root@monitor tmp]# tar xvfz /usr/local/src/nagios-plugins-1.4.16.tar.gz
```

配置 Nagios 插件一般需要 4 个选项，`--prefix` 指定插件安装路径，`--with-nagios-user` 指定插件属主，`--with-nagios-group` 指定插件的用户组，而`--enable-perl-modules` 开关允许用 Perl 语言来编写 Nagios 插件。

```
[root@monitor nagios-plugins-1.4.16]# ./configure
--prefix=/usr/local/nagios --with-nagios-user=nagios
--with-nagios-group=nagios --enable-perl-modules
```

上述 Congifure 脚本执行完毕后，可检查清单文件。


```

config.status: creating po/Makefile
--with-apt-get-command:
--with-ping6-command: /bin/ping6 -n -U -w %d -c %d %s
--with-ping-command: /bin/ping -n -U -w %d -c %d %s
--with-ipv6: yes
--with-mysql: /usr/bin/mysql_config
--with-openssl: yes
--with-gnutls: no
--enable-extra-opts: no
--with-perl: /usr/bin/perl
--enable-perl-modules: yes
--with-cgiurl: /nagios/cgi-bin
--with-trusted-path: /bin:/sbin:/usr/bin:/usr/sbin
--enable-libtap: no

```

确认配置无误，可执行 `make` 和 `make install` 命令，编译并安装 Nagios 插件到 `/usr/local/nagios/libexec` 目录。

```
[root@monitor nagios-plugins-1.4.16]# make && make install
```

Nagios 插件编译并安装完毕后，进入插件目录，执行检测插件。

```

[root@monitor src]# cd nagios-plugins-1.4.16
[root@monitor libexec]# ./check_icmp localhost
OK - localhost: rta=0.049ms, lost 0%|rta=0.049ms;200.000;500.000;0;
    pl=0%;40;80;; rtmax=0.108ms;;; rtmin=0.032ms;;;

```

如上所示，使用 `check_icmp` 命令检查本机正常在线，说明插件编译和安装正常。

5.5 配置 Nagios 的 Web 用户界面

Nagios 除了提供核心的监控功能外，还提供了一个普通的 Web 用户界面，向用户提供查看监控项状态、提交检测命令等简单的交互功能。Nagios 的 Web 用户界面大部分是 C 语言编写的 CGI 程序，需要 Apache Web 服务器来解释执行，接下来的工作是在 Apache 中配置 Nagios 相关的 CGI 目录和 Web 目录，使其能够正常执行 Nagios 的 CGI 程序，并正常显示 Nagios 的 Web 用户界面。由于 Apache 新旧版本配置文件具有一致性，下面的配置步骤在 Apache 的 1.3、2.0 以及 2.2 版本中都适合。

如表 5-2 所示，如果您配置和编译 Nagios 时并未使用 `--with-cgiurl` 开关来另外指定 Nagios CGI 的访问路径，那么 Nagios Web 界面的默认访问路径就是 `http://URL/nagios/cgi-bin`（在服务器端的真实目录就是 `/usr/local/nagios/sbin`），而相应的 HTML 文件（路径是 `http://URL/nagios`）的真实目录就是 `/usr/local/nagios/share`。

在 Nagios 3.0 版本中，使用 `make install-webconf` 命令可以配置相应的 Nagios 目录并设置

URL 别名。

```
[root@monitor nagios]# make install-webconf
```

该命令将 Nagios 相关的 Apache 配置文件 nagios.conf 安装到 Apache 的配置文件目录 /etc/httpd/conf.d（在有些 Linux 发行版中，该目录是/etc/apache2/conf.d）中。该文件内容如下。

```
# SAMPLE CONFIG SNIPPETS FOR APACHE Web SERVER
# Last Modified: 11-26-2005
#
# This file contains examples of entries that need
# to be incorporated into your Apache web server
# configuration file. Customize the paths, etc. as
# needed to fit your system.
```

```
ScriptAlias /nagios/cgi-bin "/usr/local/nagios/sbin"
```

```
<Directory "/usr/local/nagios/sbin">
```

```
# SSLRequireSSL
```

```
Options ExecCGI
```

```
AllowOverride None
```

```
Order allow,deny
```

```
Allow from all
```

```
# Order deny,allow
```

```
# Deny from all
```

```
# Allow from 127.0.0.1
```

```
AuthName "Nagios Access"
```

```
AuthType Basic
```

```
AuthUserFile /usr/local/nagios/etc/htpasswd.users
```

```
Require valid-user
```

```
</Directory>
```

```
Alias /nagios "/usr/local/nagios/share"
```

```
<Directory "/usr/local/nagios/share">
```

```
# SSLRequireSSL
```

```
Options None
```

```
AllowOverride None
```

```
Order allow,deny
```



```

Allow from all
# Order deny,allow
# Deny from all
# Allow from 127.0.0.1
AuthName "Nagios Access"
AuthType Basic
AuthUserFile /usr/local/nagios/etc/htpasswd.users
Require valid-user
</Directory>

```

在上述配置文件中,ScriptAlias 指令确保了 Apache 能够以 `http://nagios-server/nagios/cgi-bin` 的 URL 来访问 Nagios 的 CGI 目录 `/usr/local/nagios/sbin`。而 Options ExecCGI 选项确保 Apache 能够识别并执行 `/usr/local/nagios/sbin` 目录里的所有 CGI 脚本。Order 和 Allow 选项能够确定了 Apache 服务器能够接受的访问源,换句话说,如果您想限制只有个别类型的客户端能够访问 Nagios 的 URL,则需要调换 Order 参数的顺序。

```

Order deny,allow
Deny from all
Allow from 127.0.0.1
Allow from 192.68.1.0/24

```

上述例子使 Apache 上的 Nagios 服务仅接受来自 192.168.1.0 网段和本机的访问请求,其余客户端的请求全部拒绝。而剩下的 AuthName、AuthType 和 AuthUserFile 选项作为网页的认证与授权所用,在接下来的章节中我们还要继续讨论。

nagios.conf 配置文件的后半部分与前半部分雷同,是为了让 Apache 能够以 `http://nagios-server/nagios` 的 URL 来访问 Nagios 的 HTML 目录 `/usr/local/nagios/share`。

nagios.conf 配置文件基本上不需要修改,如果修改并保存,需要重启 Apache 服务和来重新装载配置文件,并启动 nagios 服务。

```

[root@monitor conf.d]# service httpd restart
停止 httpd: [确定]
正在启动 httpd: httpd: Could not reliably determine the server's fully qualified domain name, using ::1 for ServerName
[root@monitor conf.d]# service nagios start
Starting nagios: done.

```

5.6 SELinux

根据百度百科,SELinux(Security-Enhanced Linux) 是美国国家安全局 (NSA) 对于强制访问控制的实现,是 Linux 历史上最杰出的安全子系统。NSA 是在 Linux 社区的帮助下开发了一种访问控制体系,在这种访问控制体系的限制下,进程只能访问那些在他的任务中所需要文件。SELinux 默认安装在 Fedora 和 Red Hat Enterprise Linux 上,也可以作为其他发行版上容易安装的包得到。

在开启 SELinux 机制的情况下,Apache Web 服务器仅能访问到显著声明的文件,而 Nagios

的 CGI 目录/usr/local/nagios/sbin 和 HTML 目录/usr/local/nagios/share 里的文件则不在其中，后果就是操作系统拒绝 Apache 对于以上目录的访问，导致 Nagios 无法被正常访问。

使用 `getenforce` 命令可以检查系统是否开启了 SELinux 机制，而使用 `setenforce 0` 命令可以关闭该机制。

```
[root@monitor nagios]# getenforce
Enforcing
[root@monitor nagios]# setenforce 0
[root@monitor nagios]# getenforce
Permissive
```

使用 `setenforce 0` 命令仅可以临时关闭 SELinux 机制，在系统重启后会丢失更改。如果需要永久关闭该机制，则需要修改/etc/selinux/config 文件里的相关配置。

使用 `vi /etc/selinux/config` 命令编辑 SELinux 的配置文件。

```
[root@monitor /]# vi /etc/selinux/config
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#   enforcing - SELinux security policy is enforced.
#   permissive - SELinux prints warnings instead of enforcing.
#   disabled - No SELinux policy is loaded.
#SELINUX=enforcing
# SELINUXTYPE= can take one of these two values:
#   targeted - Targeted processes are protected,
#   mls - Multi Level Security protection.
#SELINUXTYPE=targeted
SELINUX=disabled
```

将选项 `SELINUX=enforcing` 和 `SELINUXTYPE=targeted` 注释掉，并新增配置项 `SELINUX=disabled`。之后使用 `vi` 的 `wq` 指令保存文件，并使用 `shutdown -r now` 重启系统。

系统重启完毕后，再次使用 `getenforce` 命令检查 SELinux 机制是否已经关闭。

```
[root@monitor ~]# getenforce
Disabled
```

5.7 访问用户认证与授权

在设计之初，Nagios 就考虑到了用户访问时的认证与授权问题。Nagios 仅允许经认证的用户访问其 CGI 目录，这就意味着只允许已登录用户访问，未登录用户是没有权限访问任何 CGI 程序的，全部会被阻挡到 Nagios 之外。

采用该认证与授权的机制是经过深思熟虑的。Nagios 除了向用户提供监控项显示及部分

HTML 文件显示外, 还允许用户通过 Web 接口向 Nagios 发送一些命令, 该功能由 Nagios 的外部访问接口提供。例如, 用户可以通过 Nagios 的 Web 界面来提交是否开启某个监控项的命令, 甚至提交重启后台 Nagios 服务的命令等, 显而易见, 只有经授权的用户才能这么做。

为了实现用户认证与授权功能, 首先需确保 `/usr/local/nagios/etc/cgi.cfg` 配置文件中的 `use_authentication` 设置为 1。

```
# AUTHENTICATION USAGE
# This option controls whether or not the CGIs will use any
# authentication when displaying host and service information, as
# well as committing commands to Nagios for processing.
#
# Read the HTML documentation to learn how the authorization works!
#
# NOTE: It is a really *bad* idea to disable authorization, unless
# you plan on removing the command CGI (cmd.cgi)! Failure to do
# so will leave you wide open to kiddies messing with Nagios and
# possibly hitting you with a denial of service attack by filling up
# your drive by continuously writing to your command file!
#
# Setting this value to 0 will cause the CGIs to *not* use
# authentication (bad idea), while any other value will make them
# use the authentication functions (the default).

use_authentication=1
```

该选项在 Nagios 安装之后即为默认开启。Apache 能够提供的最基本的认证与授权措施是基于密码文件的形式, 可在 Apache 配置目录中的 `nagios.conf` 文件 (`/etc/httpd/conf.d/nagios.conf`) 中查看。

```
<Directory "/usr/local/nagios/sbin">
# SSLRequireSSL
Options ExecCGI
AllowOverride None
Order allow,deny
Allow from all
# Order deny,allow
# Deny from all
# Allow from 127.0.0.1
AuthName "Nagios Access"
AuthType Basic
AuthUserFile /usr/local/nagios/etc/htpasswd.users
Require valid-user
</Directory>
```

上述配置中, `AuthName` 项指明了当浏览器弹出用户认证对话框时显示的标题。`AuthType` 项表明为基本的认证与授权方式, 即 Apache 会要求用户输入用户名和密码信息, 不需要加密。而 `AuthUserFile` 项指明密码文件的存放位置, 最后一项 `Require valid-user` 意味着只要是输入有效用户名和密码信息的用户都可以访问 Nagios 的 CGI 程序, 不论该用户属于哪个用户组。

而此处的访问用户可以通过 Apache 提供的 `htpasswd` 命令来产生。

```
[root@monitor etc]# htpasswd -c htpasswd.users nagios
New password:
Re-type new password:
Adding password for user nagios
```

上述命令在 `/usr/local/nagios/etc` 目录下的 `htpasswd.users` 文件中新增了一个用户名为 `nagios` 的配置项，密码经过加密。

```
[root@monitor etc]# cd /usr/local/nagios/etc
[root@monitor etc]# ls
cgi.cfg htpasswd.users nagios.cfg objects resource.cfg
[root@monitor etc]# more htpasswd.users
nagios:sk4uiQSDCCZOo
```

如果用户想继续添加用户，可再次使用 `htpasswd` 命令，只不过需要去掉 `-c` 参数，否则会覆盖原有的用户名而不是添加新用户。

```
[root@monitor etc]# htpasswd htpasswd.users nagiosadmin
New password:
Re-type new password:
Adding password for user nagiosadmin
[root@monitor etc]# more htpasswd.users
nagios:sk4uiQSDCCZOo
nagiosadmin:9x5iPsbLImtBk
```

以上命令新建了 `nagiosadmin` 用户，继续保留原有的 `nagios` 用户。为安全起见，使用 `chown` 命令只允许 `nagios` 用户访问该文件，并使用 `chmod` 命令为该文件设置读写权限。

```
[root@monitor etc]# chown nagios:nagios htpasswd.users
[root@monitor etc]# chmod g+w htpasswd.users
```

使用 `ls -altr` 命令观察 `htpasswd.users` 的属性，发现已更改为 `nagios` 用户，且读写属性已变。

```
[root@monitor etc]# ls -altr
76
-rw-rw-r--. 1 nagios nagios 44710 4月 22 21:47 nagios.cfg
-rw-rw-r--. 1 nagios nagios 11669 4月 22 21:47 cgi.cfg
-rw-rw----. 1 nagios nagios 1340 4月 22 21:47 resource.cfg
drwxrwxr-x. 2 nagios nagios 4096 4月 22 21:47 objects
drwxr-xr-x. 10 nagios nagios 4096 4月 23 15:12 ..
drwxrwxr-x. 3 nagios nagios 4096 4月 23 18:01 .
```



```
-rw-rw-r-- 1 nagios nagios 47 4月 23 18:04 httpasswd.users
```

以上配置做好之后,我们使用 `service httpd restart` 命令重启 Apache 服务,使用 `service nagios restart` 命令重启 Nagios 服务,打开浏览器,输入 `http://your-ip/nagios`,在认证与授权对话框的用户名栏内输入 `nagios`,密码栏内输入相应密码,即可看到 Nagios 页面,如图 5-3 所示,意味着 Nagios 安装并部署成功。

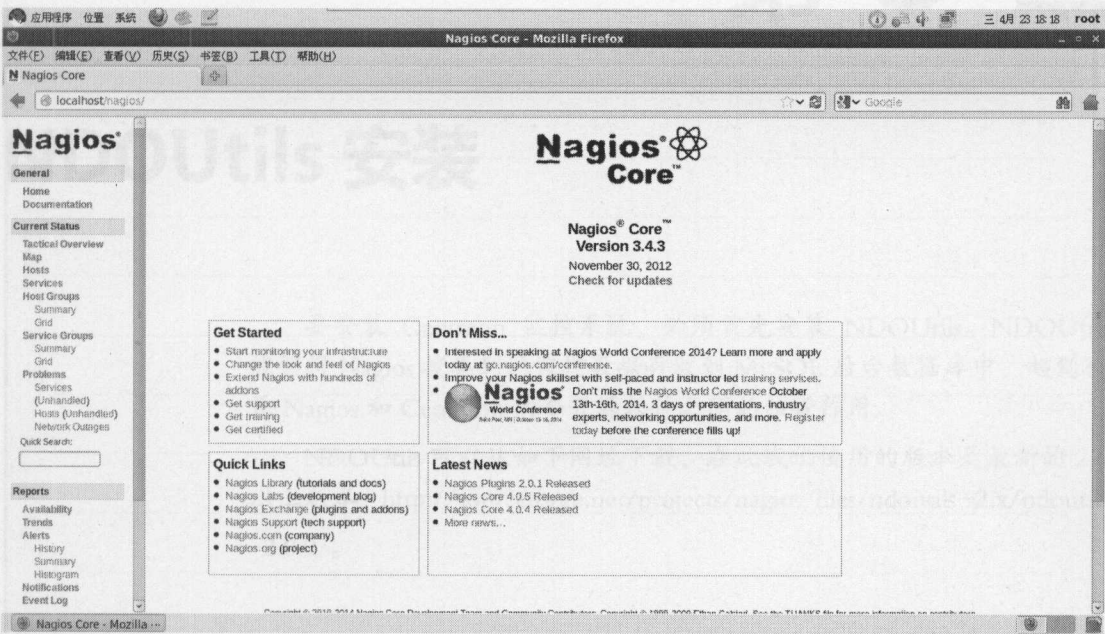


图 5-3 Nagios 默认页面



Handwritten notes area with horizontal lines.

第 6 章

NDOUTils 安装

要安装 Centreon 监控系统，必须首先安装 NDOUTils。NDOUTils 能够将 Nagios 搜集到的监控数据存放到 MySQL 后台数据库中，起到联系 Nagios 和 Centreon 两个开源软件之间的纽带作用。

NDOUTils 可以从如下网址下载，在此我们使用的版本是最新的 2.0 以上版本：<http://sourceforge.net/projects/nagios/files/ndoutils-2.x/ndoutils-2.0.0/>。

6.1 配置并编译 NDOUtils

将下载 `ndoutils-2.0.0.tar.gz` 文件上传至 Nagios 监控服务器上,解压缩后进入 `ndoutils-2.0.0` 目录,首先使用 `configure` 命令配置编译选项。

```
[root@monitor ndoutils-2.0.0]# ./configure
*** Configuration summary for ndoutils 2.0.0 02-28-2014 ***:
General Options:
-----
NDO2DB user:    nagios
NDO2DB group:   nagios
Review the options above for accuracy.  If they look okay,
type 'make' to compile the NDO utilities.
```

接下来开始编译 NDOUtils 软件。

```
[root@monitor ndoutils-2.0.0]# make
cd ./src && make
make[1]: Entering directory `/tmp/ndoutils-2.0.0/src'
gcc -fPIC -g -O2 -I/usr/include/mysql -DHAVE_CONFIG_H -c -o io.o io.c
gcc -fPIC -g -O2 -I/usr/include/mysql -DHAVE_CONFIG_H -c -o utils.o utils.c
gcc -g -O2 -I/usr/include/mysql -DHAVE_CONFIG_H -o file2sock file2sock.c
io.o utils.o -lm -lnsl
gcc -g -O2 -I/usr/include/mysql -DHAVE_CONFIG_H -o log2ndo log2ndo.c io.o
utils.o -lm -lnsl
make ndo2db-2x
make[2]: Entering directory `/tmp/ndoutils-2.0.0/src'
gcc -g -O2 -I/usr/include/mysql -DHAVE_CONFIG_H -c -o db.o db.c
gcc -g -O2 -I/usr/include/mysql -DHAVE_CONFIG_H -D BUILD_NAGIOS_2X -c -o
dbhandlers-2x.o dbhandlers.c
gcc -g -O2 -I/usr/include/mysql -DHAVE_CONFIG_H -D BUILD_NAGIOS_2X -o
ndo2db-2x queue.c ndo2db.c dbhandlers-2x.o io.o utils.o db.o -lnsl
-rdynamic -L/usr/lib64/mysql -lmysqlclient -lz -lcrypt -lnsl -lm -lssl
-lcrypto -lm
make[2]: Leaving directory `/tmp/ndoutils-2.0.0/src'
make ndo2db-3x
make[2]: Entering directory `/tmp/ndoutils-2.0.0/src'
gcc -g -O2 -I/usr/include/mysql -DHAVE_CONFIG_H -D BUILD_NAGIOS_3X -c -o
dbhandlers-3x.o dbhandlers.c
gcc -g -O2 -I/usr/include/mysql -DHAVE_CONFIG_H -D BUILD_NAGIOS_3X -o
ndo2db-3x queue.c ndo2db.c dbhandlers-3x.o io.o utils.o db.o -lnsl
-rdynamic -L/usr/lib64/mysql -lmysqlclient -lz -lcrypt -lnsl -lm -lssl
-lcrypto -lm
```

```

make[2]: Leaving directory `/tmp/ndoutils-2.0.0/src'
make ndo2db-4x
make[2]: Entering directory `/tmp/ndoutils-2.0.0/src'
gcc -g -O2 -I/usr/include/mysql -DHAVE_CONFIG_H -I ../include/nagios-4x \
-D BUILD_NAGIOS_4X -c -o dbhandlers-4x.o dbhandlers.c
gcc -g -O2 -I/usr/include/mysql -DHAVE_CONFIG_H -D BUILD_NAGIOS_4X -o \
ndo2db-4x queue.c ndo2db.c dbhandlers-4x.o io.o utils.o db.o -lnsl \
-rdynamic -L/usr/lib64/mysql -lmysqlclient -lz -lcrypt -lnsl -lm -lssl \
-lcrypto -lm
make[2]: Leaving directory `/tmp/ndoutils-2.0.0/src'
make ndomod-2x.o
make[2]: Entering directory `/tmp/ndoutils-2.0.0/src'
gcc -fPIC -g -O2 -I/usr/include/mysql -DHAVE_CONFIG_H -D BUILD_NAGIOS_2X \
-o ndomod-2x.o ndomod.c io.o utils.o -shared -lnsl
make[2]: Leaving directory `/tmp/ndoutils-2.0.0/src'
make ndomod-3x.o
make[2]: Entering directory `/tmp/ndoutils-2.0.0/src'
gcc -fPIC -g -O2 -I/usr/include/mysql -DHAVE_CONFIG_H -D BUILD_NAGIOS_3X \
-o ndomod-3x.o ndomod.c io.o utils.o -shared -lnsl
make[2]: Leaving directory `/tmp/ndoutils-2.0.0/src'
make ndomod-4x.o
make[2]: Entering directory `/tmp/ndoutils-2.0.0/src'
gcc -fPIC -g -O2 -I/usr/include/mysql -DHAVE_CONFIG_H \
-I ../include/nagios-4x -DBUILD_NAGIOS_4X -o ndomod-4x.o ndomod.c io.o \
utils.o -shared -lnsl
make[2]: Leaving directory `/tmp/ndoutils-2.0.0/src'
gcc -g -O2 -I/usr/include/mysql -DHAVE_CONFIG_H -o sockdebug sockdebug.c \
io.o utils.o -lm -lnsl
make[1]: Leaving directory `/tmp/ndoutils-2.0.0/src'

```

6.2 拷贝编译后的文件至运行目录

编译完毕的 NDOUtils 文件位于当前目录的 src 子目录下, 进入 src 子目录, 然后查看文件。

```

[root@monitor ndoutils-2.0.0]# cd src
[root@monitor src]# ls -ltr
2716
-rw-rw-r-- 1 root root 4623 3月 1 02:13 utils.c
-rw-rw-r-- 1 root root 3418 3月 1 02:13 sockdebug.c
-rw-rw-r-- 1 root root 4484 3月 1 02:13 queue.c
-rw-rw-r-- 1 root root 514 3月 1 02:13 protonum.c
-rw-rw-r-- 1 root root 169479 3月 1 02:13 ndomod.c

```



```
-rw-rw-r-- 1 root root 59500 3月 1 02:13 ndo2db.c
-rw-rw-r-- 1 root root 5453 3月 1 02:13 Makefile.in
-rw-rw-r-- 1 root root 6780 3月 1 02:13 log2ndo.c
-rw-rw-r-- 1 root root 11485 3月 1 02:13 io.c
-rw-rw-r-- 1 root root 5421 3月 1 02:13 file2sock.c
-rw-rw-r-- 1 root root 172237 3月 1 02:13 dbhandlers.c
-rw-rw-r-- 1 root root 24143 3月 1 02:13 db.c
-rw-r--r-- 1 root root 5535 5月 27 15:52 Makefile
drwxrwxr-x 8 root root 4096 5月 27 15:52 ..
-rw-r--r-- 1 root root 25560 5月 27 15:55 io.o
-rw-r--r-- 1 root root 12256 5月 27 15:55 utils.o
-rwxr-xr-x 1 root root 40017 5月 27 15:55 file2sock
-rwxr-xr-x 1 root root 41743 5月 27 15:55 log2ndo
-rw-r--r-- 1 root root 57984 5月 27 15:55 db.o
-rw-r--r-- 1 root root 218240 5月 27 15:55 dbhandlers-2x.o
-rwxr-xr-x 1 root root 284861 5月 27 15:55 ndo2db-2x
-rw-r--r-- 1 root root 218192 5月 27 15:55 dbhandlers-3x.o
-rwxr-xr-x 1 root root 284829 5月 27 15:55 ndo2db-3x
-rw-r--r-- 1 root root 219824 5月 27 15:55 dbhandlers-4x.o
-rwxr-xr-x 1 root root 285293 5月 27 15:55 ndo2db-4x
-rwxr-xr-x 1 root root 167506 5月 27 15:55 ndomod-2x.o
-rwxr-xr-x 1 root root 174856 5月 27 15:55 ndomod-3x.o
-rwxr-xr-x 1 root root 175450 5月 27 15:55 ndomod-4x.o
drwxrwxr-x 2 root root 4096 5月 27 15:55 .
-rwxr-xr-x 1 root root 36133 5月 27 15:55 sockdebug
```

如上所示, 编译 NDOUtils 后, 可以产生多个以 ndo2db-开头的文件, 其中含有 “2X” 字符串的, 是与 Nagios 2.0 以上版本兼容, 以此类推。由于本书选用的 Nagios 版本为 3.4.1, 因此适用的 NDOUtils 版本为 ndo2db-3x 以及 ndomod-3x.o。

接着我们将 NDOUtils 可执行文件手工拷贝到 Nagios 的运行目录里。

```
[root@monitor src]# cp ndomod-3x.o ndo2db-3x log2ndo file2sock
/usr/local/nagios/bin
[root@monitor src]# cd /usr/local/nagios/bin
[root@monitor bin]# ls -altr
1248
-rwxrwxr--. 1 nagios nagios 640032 4月 22 21:46 nagios
-rwxrwxr--. 1 nagios nagios 44064 4月 22 21:46 nagiosstats
-rw-rw-r--. 1 nagios nagios 31878 4月 22 21:46 pl.pl
drwxr-xr-x. 10 nagios nagios 4096 4月 23 15:12 ..
-rwxr-xr-x 1 root root 174856 5月 27 16:05 ndomod-3x.o
```



```
-rwxr-xr-x 1 root root 284829 5月 27 16:05 ndo2db-3x
-rwxr-xr-x 1 root root 41743 5月 27 16:05 log2ndo
-rwxr-xr-x 1 root root 40017 5月 27 16:05 file2sock
drwxrwxr-x. 2 nagios nagios 4096 5月 27 16:05 .
```

6.3 检查 MySQL 的配置

由于 MySQL 的默认设置不能满足 NDOUtils 组件对数据库的访问需求，因此有必要增强 MySQL 的配置，使其能够接收更多并发连接。

使用 vi 编辑 MySQL 的配置文件。

```
vi /etc/my.cnf
```

在[mysqld]下添加以下内容。

```
[mysqld]
set-variable=max_connections=1000
set-variable=max_user_connections=500
set-variable=wait_timeout=200
```

其中：

max_connections 设置最大连接数为 1000。

max_user_connections 设置每用户最大连接数为 500。

wait_timeout 表示 200 秒后将关闭空闲（IDLE）的连接，但是对正在工作的连接不影响。

如图 6-1 所示。

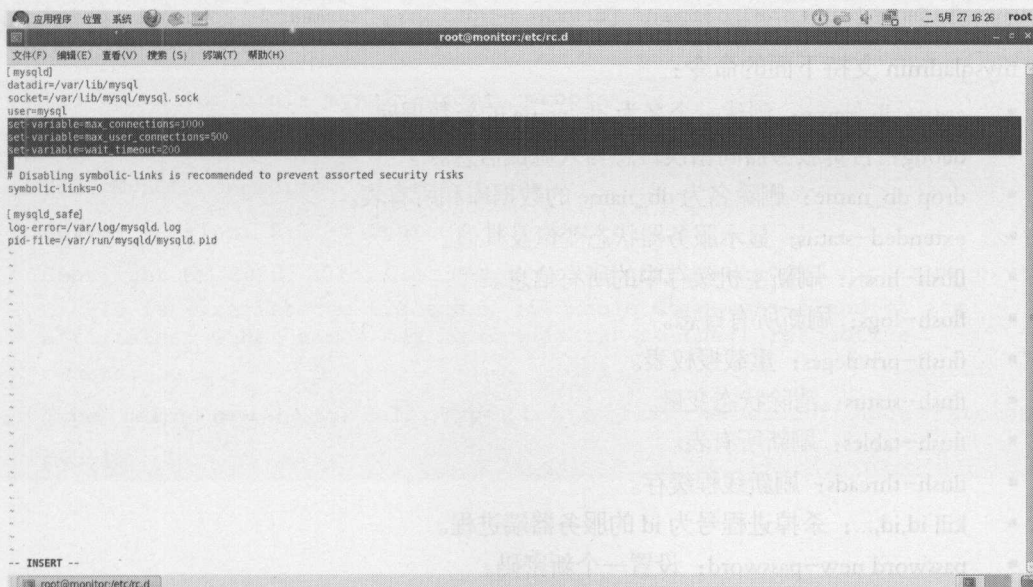


图 6-1 修改 MySQL 的配置文件

修改完毕后，保存退出，并使用如下命令启动 MySQL。

```
service mysqld start
```

接着输入如下命令登录 MySQL，检查重启后参数是否生效。

```
# mysqladmin -uroot -p variables
Password: (输入您的MySQL数据库密码)
```

看到以下项时说明参数修改成功。

```
|max_connections      | 1000
|max_user_connections | 500
|wait_timeout         | 200
```

6.4 创建 NDOUtils 数据库表

下面开始使用 MySQL 自带的管理工具 `mysqladmin` 来创建 NDOUtils 数据库。

在此，用户可以创建 NDOUtils 数据库实例，并为未来存放 Nagios 监控数据的 MySQL 数据库实例指定一个名字，可以选择例如 `nagios` 或者 `ndodb` 等方便记忆的名称。同时，您还需要为该数据库指定一个用户。

在本书中，我们选择使用 `ndodb` 作为 NDOUtils 数据库的实例名，并为方便起见，为该实例指定的数据库用户名为 `ndodb`。

`mysqladmin` 是用来执行 MySQL 数据库管理操作的客户端命令行工具，可以用来创建并删除一个 MySQL 数据库的实例，并且可以检查 MySQL 服务器的配置和当前的状态。

可以使用如下命令调用 `mysqladmin`：

```
mysqladmin [options] command [command-options] [command [command-options]] ...
```

`mysqladmin` 支持下面的命令：

- `create db_name`：创建一个名为 `db_name` 的新数据库。
- `debug`：告诉服务器向错误日志写入调试信息。
- `drop db_name`：删除名为 `db_name` 的数据库和所有表。
- `extended-status`：显示服务器状态变量及其值。
- `flush-hosts`：刷新主机缓存中的所有信息。
- `flush-logs`：刷新所有日志。
- `flush-privileges`：重载授权表。
- `flush-status`：清除状态变量。
- `flush-tables`：刷新所有表。
- `flush-threads`：刷新线程缓存。
- `kill id,id,...`：杀掉进程号为 `id` 的服务器端进程。
- `password new-password`：设置一个新密码。

将用 `mysqladmin` 连接服务器使用的账户的密码更改为 `new-password`。如果 `new-password`

包含空格或其他命令解释符的特殊字符，需要用引号将它引起来。在 Windows 中，一定要使用双引号而不要用单引号，单引号不会从密码中剥离出来，而是解释为密码的一部分。例如如下命令。

```
mysqladmin password "my new password"
```

- ping: 检查服务器是否仍活动。如果服务器在运行，则返回状态 0，如果不运行返回 1。即使出现错误例如 Access denied 也为 0，因为这说明服务器在运行但拒绝了连接，与服务器不在运行不同。
- processlist: 显示活动服务器线程的列表。类似 SHOW PROCESSLIST 语句的输出。如果给出了 --verbose 选项，输出类似 SHOW FULL PROCESSLIST。
- reload: 重载授权表。
- refresh: 刷新所有表并关闭和打开日志文件。
- shutdown: 关闭服务器。
- start-slave: 开始从服务器上的复制。
- status: 显示短服务器状态消息。
- stop-slave: 停止从服务器上的复制。
- variables: 显示服务器系统变量及其值。
- version: 显示服务器的版本信息。

在了解 mysqladmin 命令的用法之后，我们可以着手 ndodb 数据库实例的创建工作：

首先要确保 MySQL 数据库服务是启动状态，如果是停止状态，则需要启动 MySQL 数据库服务。

```
[root@monitor tmp]# service mysqld status
mysqld
[root@monitor tmp]# service mysqld start
```

登陆 MySQL 控制台。

```
[root@monitor bin]# mysql -uroot -proot
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 5
Server version: 5.1.71 Source distribution
Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql>
```


使用如下命令创建 ndodb 数据库实例。

```
mysql> create database ndodb;
Query OK, 1 row affected (0.03 sec)
```

使用 show databases 命令检查 ndodb 实例的创建结果。

```
mysql> show databases;
+-----+
| Database                |
+-----+
| information_schema      |
| mysql                   |
| ndodb                    |
| test                     |
+-----+
4 rows in set (0.00 sec)
```

结果显示 ndodb 数据库实例已成功创建。

接下来是将 ndodb 数据库用户指定为 ndodb，并为 ndodb 用户设置密码。

```
mysql> grant all on ndodb.* to ndodb@localhost Identified by "ndodb";
Query OK, 0 rows affected (0.00 sec)
```

接下来是重要的一步，使用 NDOUtils 工具提供的脚本 installdb 来初始化刚刚创建的 ndodb 数据库，创建存放 Nagios 监控数据的各类表格、索引等数据库对象。

进入 NDOUtils 存放目录的 db 子目录。

```
[root@monitor ndoutils-2.0.0]# cd db
[root@monitor db]# ls
installdb mysql-upgrade-1.3.sql mysql-upgrade-1.4b5.sql prepsql
mysql-mods-1.4b5.sql mysql-upgrade-1.4b1.sql mysql-upgrade-1.4b6.sql
queries
mysql-mods-1.4b7.sql mysql-upgrade-1.4b2.sql mysql-upgrade-1.4b8.sql
README
mysql-mods-1.4b8.sql mysql-upgrade-1.4b3.sql mysql-upgrade-2.0.0.sql
upgradedb
mysql.sql mysql-upgrade-1.4b4.sql mysql-upgrade-2.0.1.sql
[root@monitor db]# pwd
/tmp/ndoutils-2.0.0/db
[root@monitor db]# ./installdb -u ndodb -p ndodb -h localhost -d ndodb
DBD::mysql::db do failed: Table 'ndodb.nagios_dbversion' doesn't exist
at ./installdb line 51.
```

```

** Creating tables for version 2.0.1
    Using mysql.sql for installation...
** Updating table nagios_dbversion
Done!

```

至此, `ndodb` 数据库实例创建成功, 可以登录 MySQL 数据库, 用“`show tables`”命令检查 `ndodb` 中新创建的各类用于存放 Nagios 监控数据的表格:

```

[root@monitor db]# mysql -uroot -proot

Welcome to the MySQL monitor.  Commands end with ; or \g.

Your MySQL connection id is 11

Server version: 5.1.71 Source distribution

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use ndodb;

Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed

mysql> show tables;

+-----+
| Tables_in_ndodb |
+-----+
| nagios_acknowledgements |
| nagios_commands |
| nagios_commenthistory |
| nagios_comments |
| nagios_configfiles |
| nagios_configfilevariables |
| nagios_conninfo |

```


nagios_contact_addresses	
nagios_contact_notificationcommands	
nagios_contactgroup_members	
nagios_contactgroups	
nagios_contactnotificationmethods	
nagios_contactnotifications	
nagios_contacts	
nagios_contactstatus	
nagios_customvariables	
nagios_customvariablestatus	
nagios_dbversion	
nagios_downtimehistory	
nagios_eventhandlers	
nagios_externalcommands	
nagios_flappinghistory	
nagios_host_contactgroups	
nagios_host_contacts	
nagios_host_parenthosts	
nagios_hostchecks	
nagios_hostdependencies	
nagios_hostescalation_contactgroups	
nagios_hostescalation_contacts	
nagios_hostescalations	
nagios_hostgroup_members	
nagios_hostgroups	
nagios_hosts	
nagios_hoststatus	
nagios_instances	
nagios_logentries	


```

| nagios_notifications |
| nagios_objects       |
| nagios_processevents |
| nagios_programstatus |
| nagios_runtimevariables |
| nagios_scheduleddowntime |
| nagios_service_contactgroups |
| nagios_service_contacts |
| nagios_service_parents |
| nagios_servicechecks |
| nagios_servicedependencies |
| nagios_serviceescalation_contactgroups |
| nagios_serviceescalation_contacts |
| nagios_serviceescalations |
| nagios_servicegroup_members |
| nagios_servicegroups |
| nagios_services |
| nagios_servicestatus |
| nagios_statehistory |
| nagios_systemcommands |
| nagios_timedequeue |
| nagios_timedequeue |
| nagios_timeperiod_timeranges |
| nagios_timeperiods |
+-----+
60 rows in set (0.00 sec)

```

6.5 配置 NDOUtils

如果想使 NDOUtils 组件和 Nagios 工具协作顺畅，就必须分别对两者做相关配置，步骤如下。

首先进入 NDOUtils 安装目录的 config 子目录，将 NDOUtils 相关的配置文件拷贝至 nagios 的配置文件目录中。

```
[root@monitor config]# pwd
/tmp/ndoutils-2.0.0/config
[root@monitor config]# ls -altr
总用量 56
-rw-rw-r-- 1 root root 5086 3月 1 02:13 ndomod.cfg-sample.in
-rw-rw-r-- 1 root root 4831 3月 1 02:13 ndo2db.cfg-sample.in
-rw-rw-r-- 1 root root 652 3月 1 02:13 nagios.cfg.in
-rw-rw-r-- 1 root root 427 3月 1 02:13 misccommands.cfg.in
-rw----- 1 root root 4835 5月 27 15:51 ndo2db.cfg-sample
-rw----- 1 root root 5104 5月 27 15:51 ndomod.cfg-sample
-rw----- 1 root root 718 5月 27 15:51 nagios.cfg
-rw----- 1 root root 439 5月 27 15:51 misccommands.cfg
drwxrwxr-x 8 root root 4096 5月 27 15:52 ..
drwxrwxr-x 2 root root 4096 5月 27 15:52 .
[root@monitor config]# cp ndo* /usr/local/nagios/etc
[root@monitor config]# cd /usr/local/nagios/etc
[root@monitor etc]# ls -altr
总用量 108
-rw-rw-r--. 1 nagios nagios 44710 4月 22 21:47 nagios.cfg
-rw-rw-r--. 1 nagios nagios 11669 4月 22 21:47 cgi.cfg
-rw-rw----. 1 nagios nagios 1340 4月 22 21:47 resource.cfg
drwxrwxr-x. 2 nagios nagios 4096 4月 22 21:47 objects
drwxr-xr-x. 10 nagios nagios 4096 4月 23 15:12 ..
-rw-rw-r-- 1 nagios nagios 47 4月 23 18:04 htpasswd.users
-rw----- 1 root root 4835 5月 31 19:59 ndo2db.cfg-sample
-rw-r--r-- 1 root root 4831 5月 31 19:59 ndo2db.cfg-sample.in
-rw----- 1 root root 5104 5月 31 19:59 ndomod.cfg-sample
-rw-r--r-- 1 root root 5086 5月 31 19:59 ndomod.cfg-sample.in
drwxrwxr-x. 3 nagios nagios 4096 5月 31 19:59 .
```


进入 Nagios 的配置文件目录 `/usr/local/nagios/etc`，将刚拷贝进来 `ndo2db.cfg-sample` 重命名为 `ndo2db.cfg`，使用 `vi` 工具修改该文件。

```
[root@monitor config]# cd /usr/local/nagios/etc
[root@monitor etc]# mv ndo2db.cfg-sample ndo2db.cfg
[root@monitor etc]# vi ndo2db.cfg
```

如图 6-2 所示，修改 NDOUtils 配置文件 `ndo2db.cfg`，这里主要修改数据库名、用户名和密码（`db_name`、`db_user`、`db_pass`），将其设置成我们之前在安装 MySQL 时创建的数据库和用户，即 `ndodb`。其余保持默认配置即可。

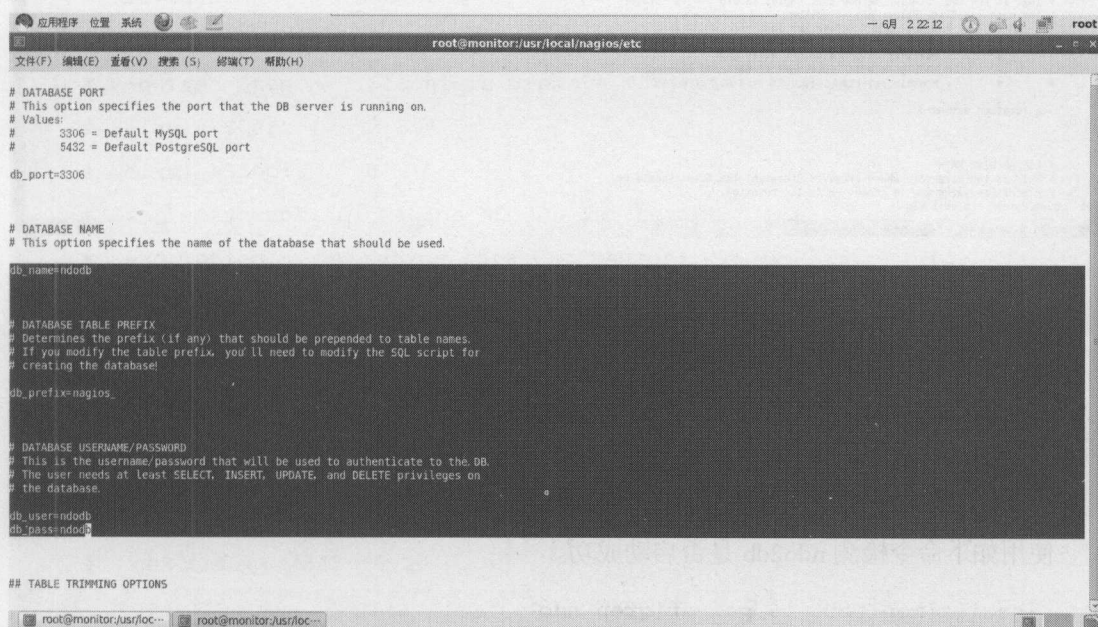


图 6-2 ndo2db 配置

接下来继续修改 Nagios 的配置文件，使 Nagios 能够正确识别并调用 NDO2db 工具。

```
vi /usr/local/nagios/etc/nagios.cfg
```

找到 `broker_module` 配置项，将下列文本配置添加到配置文件中，注意该文本为单独 1 行，不能手工换行。

```
broker_module=/usr/local/nagios/bin/ndomod-3x.o
config_file=/usr/local/nagios/etc/ndomod.cfg
```

如图 6-3 所示。


```
root@monitor:/usr/local/nagios/bin
```

```
228 #  
229 #!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!  
230 # WARNING !!! WARNING !!! WARNING !!! WARNING !!! WARNING  
231 #!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!  
232 #  
233 # Do NOT overwrite modules while they are being used by Nagios or Nagios  
234 # will crash in a fiery display of SEGFAULT glory. This is a bug/limitation  
235 # either in dlopen(), the kernel, and/or the filesystem. And maybe Nagios..  
236 #  
237 # The correct/safe way of updating a module is by using one of these methods:  
238 #   1. Shutdown Nagios, replace the module file, restart Nagios  
239 #   2. Delete the original module file, move the new module file into place, restart Nagios  
240 #  
241 # Example:  
242 #  
243 # broker_module=(modulepath) [moduleargs]  
244 #  
245 #broker_module=/somewhere/module1.o debug=0  
246 #broker_modul=/somewhere/module2.o arg1 arg2=3 debug=0  
247 #broker_modules=/usr/local/nagios/bin/ndomod-3x.x config_file=/usr/local/nagios/etc/ndomod.cfg  
248 #  
249 #  
250 # LOG ROTATION METHOD  
251 # This is the log rotation method that Nagios should use to rotate  
252 # the main log file. Values are as follows.  
253 # n = None - don't rotate the log  
254 # h = Hourly rotation (top of the hour)  
255 # d = Daily rotation (midnight every day)  
256 # w = Weekly rotation (midnight on Saturday evening)  
257 # m = Monthly rotation (midnight last day of month)  
258 #  
259 log_rotation_method=d  
260 #  
261 #  
262 # LOG ARCHIVE PATH  
263 # This is the directory where archived (rotated) log files should be  
264 # placed (assuming you've chosen to do log rotation).  
265 # quit(Enter) to exit Vim
```

```
Type: quit(Enter) to exit Vim
```

```
root@monitor:/usr/local/nagios/bin
```

图 6-3 修改 Nagios 配置文件

Nagios 配置文件修改完毕后,可以使用如下命令启动 ndo2db,但前提条件是需要确认是以 Nagios 的操作系统用户执行该命令。

```
[root@monitor etc]# su - nagios
[nagios@monitor ~]$ /usr/local/nagios/bin/ndo2db-3x -c
/usr/local/nagios/etc/ndo2db.cfg
```

使用如下命令检测 `ndo2db` 是否启动成功。

```
[nagios@monitor ~]$ ps -ef|grep ndo
nagios  18325      1  0 21:21 ?        00:00:00
           /usr/local/nagios/bin/ndo2db-3x -c /usr/local/nagios/etc/ndo2db.cfg
nagios  18328 18300  0 21:21 pts/1    00:00:00 grep ndo
[nagios@monitor ~]$ ls -altr /usr/local/nagios/var/ndo.sock
srwxrwxr-x 1 nagios nagios 0      2 21:21 /usr/local/nagios/var/ndo.sock
```

如上所示，ndo2db 启动后，会在 /usr/local/nagios/var/ 目录下产生名为 ndo.sock 的临时文件。

6.6 添加 ndo2db 为系统服务

接下来，我们需要将 `ndo2db` 的启动命令添加到系统自启动配置中。

以下命令为单独一行。

```
echo '/usr/local/nagios/bin/ndo2db-3x -c
/usr/local/nagios/etc/ndo2db.cfg' >> /etc/rc.local
```

接着，进入编译好的 NDOUtils 目录，执行下列命令。

```
# cp ./daemon-init /etc/init.d/ndo2db
# chmod +x /etc/init.d/ndo2db
```

需要注意的是，NDOUtils 工具提供的 ndo2db 启动命令存在 bug，需要使用 vi 命令编辑刚刚拷贝的/etc/init.d/ndo2db 文件，如下所示。

```
#!/bin/sh
### BEGIN INIT INFO
# Provides:          ndo2db
# Required-Start:    $local_fs $remote_fs $network $syslog
# Required-Stop:     $local_fs $remote_fs $network $syslog
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: Nagios NDO2DB Initscript
# Description:       Nagios Data Out Daemon
### END INIT INFO

#
#   Startup/shutdown script for the NDO2DB daemon under centreon/nagios.
#
#   Linux chkconfig stuff:
#
#   chkconfig: 2345 56 10
#   description: Startup/shutdown script for NDO2DB Daemon \
#               using Centreon / Nagios.

# Source function library.
. /etc/init.d/functions
# Set various vars
#DAEMON=ndo2db;
prog=ndo2db;
nagios_prefix="/usr/local/nagios"
NDODAEMON="${nagios_prefix}/bin/${prog}"
NDOCONFIG="${nagios_prefix}/etc/${prog}.cfg"
#Check if both the ndo daemon and config are found.
#echo -n "Searching ndo config & binairies : "
if test -f $NDODAEMON
then
```



```

if test -f $NDOCONFIG
then
    RETVAL=0;
    #echo "Files found!";
else
    echo "ERROR: Config file not found!";
    exit 1;
fi
else
    echo "ERROR: ndo2db not found!";
    exit 1;
fi

start () {
    if test -f "/var/lock/subsys/ndo2db"
    then
        echo "ndo2db is already running...";
        return 1;
    fi

    echo -n "Starting $prog: "
    # start daemon
    daemon $NDODAEMON -c $NDOCONFIG
    RETVAL=$?
    echo
    [ $RETVAL = 0 ] && touch /var/lock/subsys/ndo2db
    return $RETVAL
}

stop () {
    # stop daemon
    echo -n "Stopping $prog: "
    killproc $NDODAEMON
    RETVAL=$?
    echo
    [ $RETVAL = 0 ] && rm -f /var/lock/subsys/ndo2db
}

restart() {

```



```

        stop
        start
    }

    case $1 in
        start)
            start
            ;;
        stop)
            stop
            ;;
        restart)
            restart
            ;;
        condrestart)
            [ -f /var/lock/subsys/ndo2db ] && restart || :
            ;;
        reload)
            echo -n "Reloading $prog: "
            killproc $NDODAEMON -HUP
            RETVAL=$?
            echo
            ;;
        status)
            status $NDODAEMON
            RETVAL=$?
            ;;
        *)
            echo $"Usage: $prog
{start|stop|restart|condrestart|reload|status}"
            exit 3
    esac
    exit $RETVAL

```

至此, NDOUtils 工具已经安装并配置完毕, 使用如图 6-4 所示命令检查 NDOUtils 启动和停止。

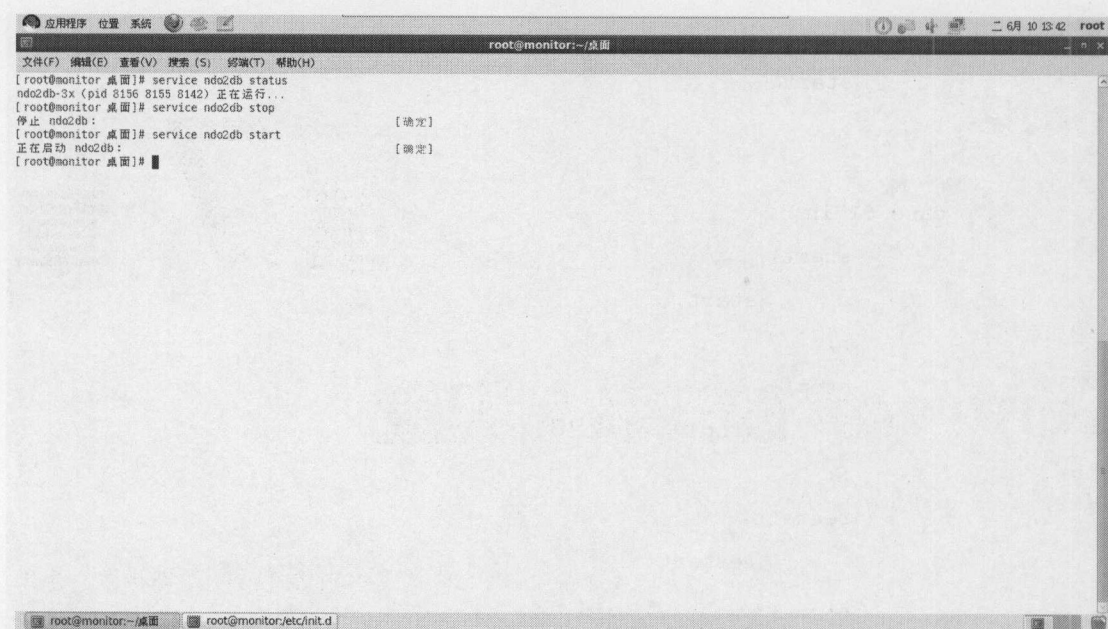


图 6-4 NDOUtils 脚本启停测试

第7章

Centreon 的安装与配置

很多人认为，IT 运维监控的过程就是检测信息系统是否异常，并就异常情况及时告警的过程。但这一点认识并不足以反映现代信息管理系统手段的全部，尤其是当 IT 运维管理工作引入了大数据的概念，并且具备预测系统告警趋势、精确分析事件持续全过程、产生精准的问题报告等等能力之后。

IT 运维监控过程存在如下关键点：

(1) 对信息系统运转过程中产生的问题迅速报警，这也是对于监控系统最基本的要求。告警速度是信息系统服务质量的重要衡量指标，一般来说，我们可以采用邮件、短信、声音以及可视化地图等形式迅速向相关人员传递各类告警信息，但一定要注意设置合适的阈值，并经常根据系统的实际运行情况进行微调，否则会被大量无效的告警消息所淹没。

(2) 不断丰富 IT 监控资源。丰富的告警信息、翔实的历史告警数据有助于系统管理员轻松预测硬盘容量增长的趋势、或者下一次服务器电源更换的时间等。随着监控项的不断增多，用户会发现对于所掌管服务器的健康状况变得越来越可预测。

(3) 要准确分析并挖掘出问题的原因。积累的历史监控数据以及监控项之间的关联特性有助于判断问题产生的原因，并且深入分析被监控对象的行为。

(4) 了解服务指标 (Service Index, SI) 的实际运作过程，且通过监控系统不断验证服务指标的有效性。监控质量及告警的及时程度在制订 IT 服务质量，以及服务响应级别 (SLA, Service Level Agreement) 相关协议时是重要的参考指标。

当然，光有 Nagios 还不足以使我们的监控平台具备上述特性，只有加上 Centreon，整个 IT 运维监控系统才能基本具备以上特点。

7.1 什么是监控以及如何监控

7.1.1 监控已经不再局限于基础设施

就本书所述的企业级 IT 运维监控系统而言，已经不仅仅局限于对于 IT 基础设施，例如服务器、网络、防火墙等对象的监控了。各种复杂的企业对象，例如业务应用、服务水平协议中定义的各项 IT 服务指标，以及整个业务流程，都已经变成运维监控系统的监控对象。

提起运维监控系统所使用的检测手段，无论哪个领域，基本的检测手段都是相似的：

- (1) 可用性检测。例如，为了检测一台服务器、一组交换机，或是应用程序开放的端口，人们经常使用 ping 命令或者 telnet 命令来实现。更进一步，人们在使用 ping 命令进行服务器可用性检测时，往往会接着进行性能检测，例如会搜集命令返回的数值，以检测网络的延迟（Latency）。
- (2) 性能检测。性能检测往往会记录一组数值，并与相关项的阈值进行比较。例如，记录 CPU 的占用率、磁盘的读写繁忙程度、或者进程的数量等等
- (3) 完整性检测。该类别的测试时为了验证某些监控项是否完整合规以及是否符合业务逻辑。例如，日志文件中的关键字检测、某一大小不符合常规的临时文件、最近一段时间的交易数量、两个系统之间业务数据的不一致等等。以上列表远远不够，因为每一个公司都有自己的监控性能指标。甚至更进一步，完整性检测可聚合多个独立的监控指标形成一个集合，以全面度量和预警公司的某类业务流程，例如监控某一业务系统的整体可用性。

7.1.2 基础设施监控

IT 基础设施的监控是迄今为止运维监控系统运用最为广泛的地方，它涵盖了由服务器和中间件组成的复杂网络内部所有的硬件以及软件层面的监控。以下是对 IT 基础设施监控范围的一些举例：

- 网络监控；
- 路由器和交换机可用性监控；
- 网络连接延迟时间及误码率监控；
- 带宽监控；
- 路由器协议一致性监控及 VLAN 监控；
- IT 基础设施监控；
- 机房温度及湿度监控（借助合适的传感器）；
- 磁盘及网卡 IO 吞吐量监控；
- 备份磁带库监控；
- RAID 状态、存储状态、冗余电源状态监控等等；
- 系统监控；
- 系统的文件系统、CPU、内存使用率监控；
- 系统日志监控；

- 系统换页空间监控等；
- 中间件及应用程序、业务监控；
- 进程监控；
- 执行一次简单业务查询的速度监控；
- 虚拟环境下的虚拟机数量及状态监控等等。

7.1.3 应用程序监控

应用程序监控的最终目的是确保应用程序正常运行，从而确保业务的连续性。在很多大型企业中，仅仅监控应用程序的端口状态、应用进程状态或者线程状态是远远不够的，尽管这些监控某些情况下有效果。除此之外，还有必要按照应用程序实际运作的那样，由相关监控项不断执行相应的应用程序功能测试，通过模拟实现应用程序可用性监控。

一般来说，需要为执行应用程序监控任务而设计专门的检测项，该类检测项会执行特定的应用场景，并检测场景输出值是否与预期值一致，例如：为了检测供应链系统是否可靠，专门设计订单创建以及取消检测项。另一方面，如果应用程序在出厂时已经由厂商内置了功能检测插件，那我们在设计相关功能检测项的时候，只需要调用该插件即可，省去了设计检测项逻辑的步骤，更加方便。

需要指出的是，安全问题应该引起特别注意，要为应用程序监控项创建专门的用户，且配置有限的权限。这样一来，监控系统所做的模拟操作才不会与正式用户所做的相混淆，并且便于审计。

7.1.4 SLA 监控

SLA 服务水平协议 (SLA, Service Level Agreement) 是在一定开销下为保障服务的性能和可靠性，服务提供商与用户间定义的一种双方认可的协定。此开销通常是驱动提供服务质量的主要因素。一个完整的 SLA 同时也是一个合法的文档，包括所涉及的当事人、协定条款(包含应用程序和支持的服务)、违约的处罚、费用和仲裁机构、政策、修改条款、报告形式和双方的义务等，同样服务提供商也可以对用户在工作负荷和资源使用方面进行规定。

传统上，SLA 包含了对服务有效性的保障，譬如对故障解决时间、服务超时等的保证。但是随着更多的商业应用在互联网上的广泛开展，越来越需要 SLA 对性能（如响应时间）作出保障。实际上，SLA 的保障是以一系列的服务水平目标 (SLO, Service Level Objects) 的形式定义的。服务水平目标是一个或多个有限定的服务组件的测量的组合。一个 SLO 被实现是指这些组件的测量值在限定范围里。SLO 有所谓的操作时段，在这个时间范围内，SLO 必须被实现。但是由于互联网的统计特性，不可能任何时候都能实现这些保障。因此 SLA 一般都有实现时间段和实现比例，实现比例被定义为 SLA 必须实现的时间与实现时段的比值。

例如：在“工作负荷<100 事务数/秒”前提下，早上 8 点到下午 5 点服务响应时间<85ms，服务有效率>95%，在一个月内的总体实现比例>97%。

根据上述定义，在本书的 IT 运维监控系统中，对于 SLA 的监控，被分解成为对于具备阈值的一些控制点（服务水平目标）的监控，只要这些控制点不超过阈值，那么 SLA 就认为

是正常的。例如，某个电商网站的可用性 SLA 监控可以被分解为以下关键控制点（SLO）的监控：

- （1）平均页面访问时间不超过 2 秒；
- （2）主页面显示正常；
- （3）联系人页面显示正常；
- （4）订单流程正常。

以上 4 个关键控制点组成了该电子商务网站的可用性 SLA，如果这些关键控制点正常，那么该购物网站的可用性 SLA 就是正常的。

7.1.5 业务活动监控

业务活动监控（business activity monitoring, BAM）这个术语是在 2002 年由咨询公司 Gartner Group 提出的，是基于企业应用集成的，用于监控企业运营状况的软件技术。它提供对业务绩效指标的实时访问，以改进业务运作的速度和效率。BAM 用于描述一些新兴的能力，这些能力将一些关键技术集中起来，从根本上改变业务系统的状况。

BAM 是应用集成技术中发展最为快速、对业务进行优化最有效的手段，其宗旨在于实时获得业务流程运行的状态，自动提供客观分析报告，以改进、优化业务流程，其改进包括技术层面，也包括人员、管理层面。业务活动监控的目标是提供当企业的业务环境发生变化时能够及时了解业务事件的能力，这样就能做出及时的决策。通过提供实时的信息，BAM 方案可以减少成本和加速执行企业事务。BAM 通过采集业务流程运行的实时信息，调用 BPM 对业务流程进行管理，使企业具备了敏捷型企业所要求的素质，能够快速地响应市场变化，快速地调整业务策略，快速地实施业务流程，同时根据反馈的信息对业务流程进行快速地优化调整。

在我们的运维监控系统中，对于业务活动或者业务流程的监控同样是通过在流程中设置关键监控点（SLO 控制点）和指标项来完成的，最好的表现形式是在一张反映业务流程的大图上进行关键节点的展示。

例如，同样是对电商网站的监控，业务活动监控主要从以下方面着力：

- （1）网站可用性和每个页面的可用性检测（适用于 7.1.4 节提到的 SLA 监控）；
- （2）网站的订单数量检测；
- （3）通过网站接收的订单数量与后台 ERP 系统订单数量的对比检测；
- （4）处于“准备”阶段的订单数量；
- （5）订单的平均等待时间和每个时段的等待数量等等。

7.2 究竟什么是运维监控

7.2.1 运维监控的原则

一般来说，IT 运维监控系统有两项基本原则：

- (1) 对被监控的元素要有最小的干扰；
- (2) 确保被监控元素之间具备最大独立性。

设想我们将运维监控系统部署在虚拟机环境下，该监控系统除了负责监控企业内部其他 IT 基础设施外，还同时负责虚拟机自身的性能监控和业务监控。从运维监控系统管理员的角度看来，自然是在虚拟机上配置的监控项越多越好，但监控项的增加可能会导致虚拟机性能的下降，意味着监控系统干扰了自身的性能。因此，从安全和影响范围的角度出发，最好避免尽可能多地在客户端安装脚本和影响性能的各种插件，要尽量从监控软件端搜集被监控端的各类性能信息和故障信息。

基于以上原则，Centreon 主要使用两类运维监控模式：主动模式和被动模式。相关内容将在下面的段落中详述。

7.2.2 主动监控模式

主动监控模式是最为经典的，它由发送查询请求，接收并分析查询结果构成。主动监控模式一般遵循以下 3 个步骤：

- (1) 监控服务器向被监控端发送请求；
- (2) 被监控端返回监控查询结果；
- (3) 监控服务器分析查询结果、判断并显示被监控项的状态。

主动监控模式是运用最为广泛的模式，采用轮询以及请求—响应的机制，具备很高的可靠性，在常见的 IT 运维监控系统中，有两项主流技术是基于主动监控模式的：

SNMP (Simple Network Management Protocol, 简单网络管理协议)，是采用主动监控模式的标准技术，也是应用最为广泛的监控技术。

WMI (Windows Management Instrumentation, Windows 管理规范)，是 Windows 平台核心的管理技术，用户可以使用 WMI 管理本地和远程计算机。

以上两类协议是运维监控系统的首选，因为它们符合监控系统应遵循的第一项原则，属于操作系统或者设备自身原生 (Native) 支持的监控协议，对于系统自身运行有最小的干扰程度。然而，对于专有系统、需要经特殊逻辑执行监控的系统，以及不支持通过 SNMP 以及 WMI 等非侵入式监控的系统而言，Nagios 还提供了可以安装在这些系统上的客户端监控代理——NRPE (Nagios Remote Plugin Executor, 即 Nagios 远程插件执行器) 来执行监控。顾名思义，NRPE 是用于在远端服务器上运行检测命令的守护进程，它接收 Nagios 监控端检测命令的驱动，调用检测插件执行检测，并返回检测结果。同时，通过 Centreon 也能够以更便利的方式管理 NRPE。

除了以上方式之外，还可以通过一些标准的硬件、软件管理协议监控相应的硬件和软件。例如，通过 IPMI (Intelligent Platform Management Interface, 智能平台管理界面) 协议来管理硬件设备，通过 JMX (Java Management Extensions, 即 Java 管理扩展) 来管理基于 Java 开发或者基于 Java 封装的程序、系统、设备等等。还有基于 SSH 或者 Telnet 协议的监控，但与上面介绍的各类监控方式相比，基于 SSH 或者 Telnet 方式的监控会显著提高监控端以及被监控端的 CPU 负载，不建议大规模使用。

7.2.3 被动监控模式

与主动监控模式不同的是，基于被动监控模式构建的监控服务器从不主动发起对被监控端的查询，而是化主动为被动，等待各个监控端主动上报自己的状态。与主动监控的服务器相比，基于被动模式的服务器不发起轮询请求，更少耗费自身资源，往往能够管理更大规模的 IT 基础设施和网络系统。

另外，很多设备部署在企业内部网的 DMZ（Demilitarized Zone，称为“隔离区”，也称“非军事化区”。它是为了解决安装防火墙后外部网络不能访问内部网络服务器的问题，而设立的一个非安全系统与安全系统之间的缓冲区）区，受到防火墙及路由器等安全设备的严密保护。如果不想调整防火墙策略，并且想要部署此类设备监控的话，采用被动模式监控机制，由设备直接向中央监控服务器发送设备健康信息或者告警信息也是可行的手段。

在被动模式下，省去了监控服务器发起轮询、等待并获取被监控项轮询结果这一流程，以及期间的时间延迟等开销，可以达到近乎实时的监控效果。如果在 Centreon 中部署了采用被动模式的被监控机，任何监控项一旦出现问题，几乎可以立即反映在 Centreon 的报警界面上，相比而言存在速度上的优势。

采用被动模式的监控系统唯一缺陷就是可能存在告警信息不及时的问题，一旦监控项出现问题，无法向外发送最新的告警信息，那么监控中心就有可能仍然显示为设备正常，但其实已经影响到业务运行了。

7.3 SNMP

Centreon 中采用最广泛的基于被动模式的监控方式是通过接收 SNMP trap 的方式来监控设备状态。

SNMP(简单网络管理协议)是一种应用层协议，是 TCP/IP 协议族的一部分。它使网络设备之间能够方便地交换管理信息。能够让网络管理员管理网络的性能，发现和解决网络问题及进行网络的扩充。目前 SNMP 已成为网络管理领域中事实上的工业标准，并被广泛支持和应用，大多数网络管理系统和平台都是基于 SNMP 的。如果监控平台需要查询被管理设备的状态，则需要通过 SNMP 的 get 操作获得设备的状态信息；同样，如果需要修改或者配置被管理设备的参数，则需要通过 SNMP 的 set 操作来完成。

MIB

MIB 是描述被管理设备上的参数的数据结构。由于通过 SNMP 方式管理的设备相当复杂，拥有很多可以被管理的参数，需要对它们进行归类、分级。管理信息库（MIB）是一个具有分层特性的信息的集合，我们可以通过 SNMP 的命令去存取它。MIB 的成员是一些被管理的对象（Managed Object），以对象标示符（Object Identifiers）来区分它们。被管理的对象由一个或多个对象实例（Object Instances）组成，本质上，这些对象实例就是变量。

在 MIB 的层次结构中，一个对象标示符唯一标识了被管理对象。MIB 的层次结构可以被描述成无根名的树，树的级别被不同的组织所划分，如图 7-1 所示。

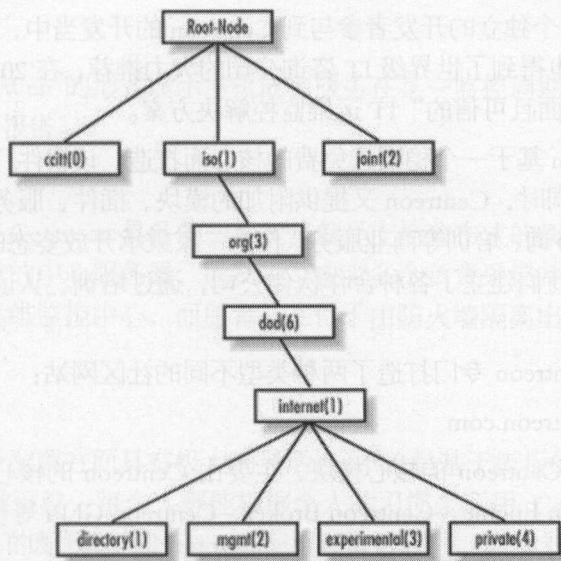


图 7-1 MIB 树形结构

很多能够被 SNMP 管理的对象都是由标准组织定义好的。比如系统磁盘的信息，其 OID 信息用“1.3.6.1.4.1.2021.9”表示。这串数字是国际标准化组织协商定义好的，大家都要去遵循它。当然，国际组织不可能预知未来，如果您要开发的设备有一些管理需求没有任何 RFC 定义过，那么您也可以编写自己的 MIB 文件来定义私有的 MIB 对象。

SNMP trap

SNMP trap 是 SNMP 的一部分，当被监控端出现特定事件，可能是性能问题，甚至是网络设备接口异常等，位于被监控端的 SNMP 客户端代理（Agent）会给监控平台发告警事件，并不需要监控平台主动轮询时才发出告警。在大规模复杂系统中，主动上报是非常有效的通知方式，正如计算机的设计者们用中断通知 CPU，让繁忙的 CPU 主动处理各类关键任务，而非主动轮询一样。在大规模监控系统中，SNMP trap 通知是合理的选择。一句话来说，SNMP trap 就是被管理设备主动发送消息给监控平台的一种机制，是被动监控模式的典型应用。

同样地，由 Nagios 提供的组件——NSCA（Nagios Service Check Acceptor，Nagios 服务检测接收器），也是基于被动监控模式的监控组件，在接下来的章节中我们会加以介绍。

7.4 Centreon——不仅仅是包装后的 Nagios

7.4.1 MERETHIS 公司简介

Centreon（<http://www.centreon.com/>）是由法国的 MERETHIS 公司开发并负责维护的一个商业项目，同时有开源的社区版本。Centreon 的商业版本和开源版本是同一套代码，通过插件可扩展多种功能。MERETHIS 公司在向开源社区不断提交新版 Centreon 核心模块代码的同时，会开发多种商业版本的插件，如知识库插件、业务流程监控插件、BI 插件等，并提供付费服务。

发展到现在，Centreon 每月有超过 12000 份的下载量，在世界范围内已经具备了 45000

个独立用户。有至少 90 个独立的开发者参与到 Centreon 的开发当中，持续为 Centreon 的发展添砖加瓦。Centreon 也得到了世界级 IT 咨询公司的大力推荐，在 2011 年，Gartner 咨询公司承认 Centreon 为“全面且可信的”IT 运维监控解决方案。

如上所述，Centreon 基于一个稳固且免费的核心而打造，该软件几乎所有的功能都是免费提供且开放源代码，同时，Centreon 又提供附加的模块、插件、服务等产品供商业客户购买，并提供技术支持、咨询、培训等商业服务。作为一家秉承开放姿态的商业公司，Centreon 的所有者还积极与集成商们建立了各种合作伙伴公司，通过培训、认证、系统集成等方式共同推广 Centreon。

基于以上模式，Centreon 专门打造了两种类型不同的社区网站：

(1) <http://forge.centreon.com>

该社区网站包含了 Centreon 的核心模块，主要由 Centreon 的核心开发者提供，包括：Centreon-core、Centreon Engine、Centreon Broker、Centreon GLPI 等模块。

(2) <http://community.centreon.com>

该社区网站主要提供开发者们提供的各类 Centreon 插件，这些插件不是 Centreon 官方提供的模块，而是志愿者们根据需要开发并免费提供的。

在上述两个社区网站中均无法下载 Centreon 的商业模块，因为这是 MERETHIS 公司的商业客户专享的。本书主要针对上述网站中能够下载并部署的开源模块（这已经包含了大部分的 Centreon 的功能），并且在有必要的时候提及部分增值的商业模块。

7.4.2 Centreon 的功能

从 Nagios 配置界面开始发展的 Centreon 已经逐渐演变成专业的 IT 运维监控解决方案，具有自身的特点：

实时监控

实时监控是 Centreon 的主要特性。实时监控的目的首先是检测事件，然后根据预先设定的阈值判断事件的级别，并在合适的时间通知合适的人。Centreon 也能够结合 Nagios 的事件升级上报 (Notification Escalations) 机制，通过预先设定的通知周期和标准来发送不同的告警信息。

除了事件通知外，Centreon 还提供了多种图形化的模型和聚合视图来表示警报的不同需求，并且实现了对于 SNMP trap 的管理，包括基于正则表达式过滤的各类陷阱信息。

需要注意的是，此处提到的“实时监控”并非真正的实时报警，多数是“早期预警”。此类报警的周期依赖于 Nagios 的轮询周期，一般为 5 分钟以内。只有真正的基于“被动监控模式”的监控项才能在故障发生的第一时刻告警，是真正的“实时监控”。

性能监控

Centreon 从 Nagios 采集到的实时监控数据库中抽取各类信息并解析，以图形和日志的形式展现出来。这些可视化数据可以用来从历史角度分析故障产生的原因，或者预测监控项的趋势，以提前采取干预措施。

监控报告

Centreon 能够以 Web 的形式展示主机或者服务在某一监控周期内的告警报表信息，支持导出 csv 形式的监控报告。

分布式架构

与竞争对手相比，Centreon 具备的一项优势就是支持分布式的架构，易于部署。该架构包括一个负责图形展示的中心服务器，以及一个或者多个负责采集并加工监控数据的卫星服务器。前者位于 IT 运维监控中心，而后者往往位于由防火墙隔离出的企业远程业务网络或者 DMZ 区内。

灵活的配置

Centreon 在管理及配置方面具有极大的灵活性，不论是基于模板的监控主机大批量部署，还是个别监控项的细微调整，每个人都能根据个人的习惯来运用 Centreon。出于和 Nagios 的深远关系，Centreon 的配置项与 Nagios 的配置项大致相同，同样具备如下概念：模型、主机、服务、通知等等，但 Centreon 也提出了自己专有的配置项，如服务模型之间的关联关系、主机模型之间的关联关系，以及发展出的一套基于用户—资源—权限模型的 ACL（Access Control List）的认证授权与管理模式等，这些都是 Nagios 不具备的。

可扩展及可集成性

和 Nagios 一样，Centreon 也是按照模块化的结构设计和实现的，这意味着它具备和 Nagios 一样的可扩展及可集成性。当你搜索 Internet 上关于 Centreon 的知识时，会发现很多功能模块已经存在，且源源不断地被热情的开发者发布出来。此外，官方的模块也可以在 <http://forge.centreon.com> 和 <http://community.centreon.com> 这两个开发者社区网站中找到。

在检测机制方面，Centreon 能够充分利用 Nagios 业已存在且不断涌现的各类监控插件来扩展自己的监控能力，并时刻受益于 Nagios 所拥有的一切开源社区。在著名的 Nagios 插件网站 exchange.nagios.org 和 monitoringexchange.org 上找到的一切检测插件，都可以被 Centreon 所利用。

在与异构系统集成方面，Centreon 同样表现出色，提供了一系列机制与其他系统进行集成。Centreon 的第一种集成方式来自于 Nagios，即通过自定义脚本的方式，发送邮件或者 HTTP 请求至外部的事件管理平台里，使得自身成为一个完整的事件管理平台的一部分。或者是相反，Centreon 的第二种集成方式是可以管理外部的事件管理平台，比如以被动监控模式集成 SNMP trap 信息、或者 Syslog 信息等。如此一来，Centreon 就可以代理并处理来自于其他监控平台的检测事件信息和通知消息，以多种技术手段实现全面监控。

相较于其他 IT 运维监控系统而言，Centreon 具备三项绝对优势。

首先，Centreon 是一个通用的开放式监控平台，且具备集成大型系统的能力。在分布式集成架构中，以 Centreon 为基础，很容易就建立起中心式的管理与展示平台，同时纳入多种业务及网络监控系统。这种灵活性使 Centreon 成为在大型异构式网络内采用量身定做的方式，构建自定义的、兼容遗留系统的、企业级的 IT 运维监控平台成为可能。

其次，Centreon 提供了诸多丰富的特性——权限管理、与 LDAP 服务器和 AD 目录服务器的同步、分布式架构、图形化展示地理信息或者业务流程、报表支持、数据库支持等，是

搭建其企业级的运维监控平台的必要条件。

最后，得益于开放式的架构和活跃的开发社区及团队，Centreon 和 Nagios 一起共享丰富多样的监控插件，很大程度上拓展了 Centreon 的功能和适用范围。

7.5 Centreon 的架构

7.5.1 系统组件

Centreon 主要包括如表 7-1 所示的组件。

- (1) 基于 Apache 的 Web 界面；
- (2) MySQL 关系型数据库；
- (3) 专门用来存储并绘制图形的 RRD 数据库（Round Robin Database，是一个强大的绘图引擎，很多工具例如 MRTG 都可以调用 rrdtool 绘图。所谓的“Round Robin”其实是一种存储数据的方式，使用固定大小的空间来存储数据，并有一个指针指向最新的数据的位置）；
- (4) 调度进程；
- (5) 代理进程；
- (6) 检测命令及探针；
- (7) 一系列后台服务和定时任务等。

以上组件均可以部署在 Linux 的操作系统中，不同的组件各有自己需要依赖的软件，在安装的时候需要特别注意，如果遇到某项软件缺失，可根据给出的提示下载适合操作系统版本的软件并安装，否则缺失软件的组件工作起来可能不正常。

表 7-1 Centreon 的组件堆栈

Centreon 的 Web 界面			
Apache 服务器			
	RRD 数据库	MySQL 数据库	
后台服务和计划任务 (centCore, centStorage, centTrapHandler..)			代理进程 (NDOUtils, Centreon Broker)
			调度进程 (Nagios, Centreon Engine...)
			检测命令及探针
基于 GNU/Linux 或类 UNIX 的操作系统			

在管理 Centreon 和 Nagios 的日常工作中，绝大部分时间都花费在 Centreon 的 Web 界面中，配置、管理、实时监控、分析等等，只有当安装新的检测命令或者探针的时候，才会登录到被监控的服务器上进行操作。

表 7-2 是安装 Centreon 版本 2.3.8 所需要的软件列表，更详细的以及最新的清单请到 Centreon 的官方网站上查找。

表 7-2 安装 Centreon 所需的软件及版本号

所需软件	支持的软件版本
操作系统	GNU/Linux x86、x86_64，或者运行在 X86 CPU 上的 Solaris 操作系统
Web 服务器	Apache 1.3, 2.0, 2.2
浏览器	Microsoft IE 7 及以上版本、Mozilla Firefox 2.0 及以上版本、Apple Safari 3 及以上版本、Google Chrome
PERL 解释器	5.8 及以上版本
PHP	PHP 5.0–5.3 Extensions: XMLWriter, GD, MySQL, LDAP（可选），SNMP, SOAP PHP 扩展库： DB, DB_DataObject, DB_DataObject_FormBuilder, MDB2, Date, HTML_Common, HTML_QuickForm, HTML_QuickForm_advmultiselect, HTML_Table, Archive_Tar, Auth_SASL, Console_Getopt, Net_Socket, Net_Traceroute, Net_Ping, Validate, XML_RPC, SOAP
数据库	MySQL 5.x InnoDB
调度及检测进程	Nagios 2 及 Nagios 3 (建议版本为 3.2.3 及以上版本) Icinga Shinken Centreon Engine
代理进程	NDOUtils Centreon Broker

7.5.2 数据存储

与 Nagios 一样，Centreon 也会创建并使用自己的后台 MySQL 数据库示例。Centreon 将所有数据存放在以下 3 类不同用途的 MySQL 数据库中：

- **centreon**：存储配置相关的信息。
Centreon 监控系统的后台调度进程如 Nagios、Icinga、Shinken 等采用文本配置文件的方式存储配置信息，centreon 数据库中的配置信息被用来生成这些配置文件。
- **centreon_status**：存储监控系统后台检测进程搜集到的，实时的告警信息和性能信息。

该数据库存放的数据是临时的，Centreon 监控系统用这些数据在 Web 界面上显示文本或者图形信息。该数据库中的数据由代理进程 NDOUtils 或者 Centreon Broker 填充，且能够被定时清除，不存放任何历史数据或者过期数据。

- **centreon_storage**：该数据库中存放监控系统的日志信息和性能数据（用以产生主机或者监控项性能趋势图）。

该数据库中由于存放历史日志以及性能信息，容量较大，其中存储的数据用以生成被监控系统的各类报表以及性能趋势图形。

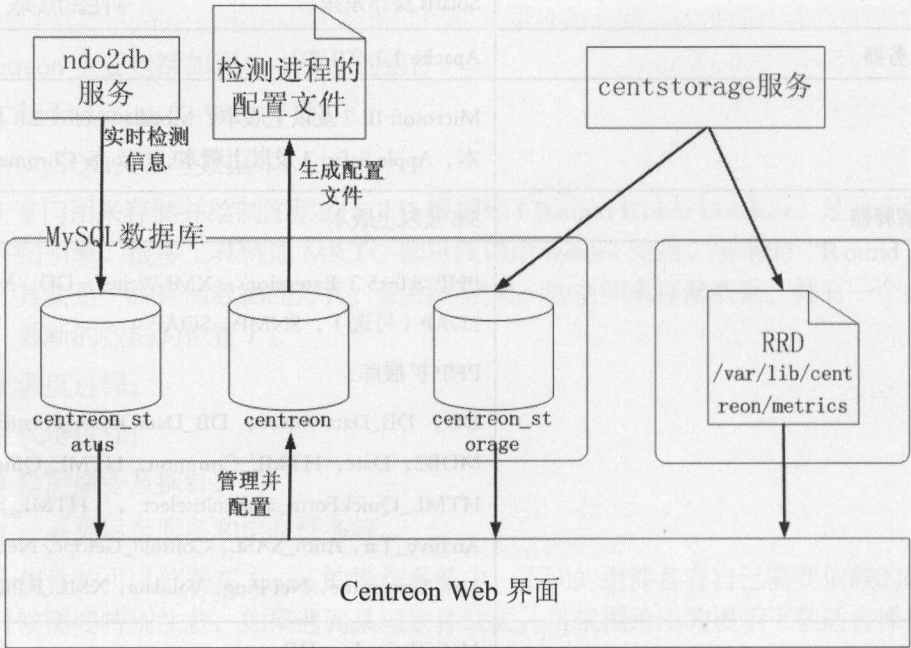


图 7-2 Centreon 的数据库相关结构

如图 7-2 所示，RRD 图形数据库中的数据文件是从 **centreon_storage** 数据库中抽取并产生的。Centreon 监控系统中漂亮的性能图片是 PNG 格式，由 RRDTools 通过 RRD 数据库中的 RRD 数据文件产生，而这些数据文件又随时可以通过 **centreon_storage** 数据库而生成。因此在 Centreon 监控系统中，**centreon_storage** 数据库起到一个存储被监控主机或者服务的性能数据的作用。

7.5.3 检测命令

Centreon 上的检测命令（**check command**）与 Nagios 中的插件（请参考 5.3 节）的承担的任务有所不同。从被监控端的角度来说，插件主要在被监控端运行，接收 Nagios 调度进程的轮询，在被监控端执行各类检查，并向 Nagios 返回告警信息和性能信息。而检测命令的概念是从监控服务器的角度来说的。具体来说，检测命令部署并运行在 Centreon 和 Nagios 监控服务器上，可以通过直接检测被监控端、或者调用被监控端的检测插件等方式来获取被监控端的告警信息或（及）性能数据。与检测插件一样，检测命令通常是由 Perl 语言开发，也可以用 C、Python、Ruby 等语言开发。

检测命令通常带有被监控服务器的 IP 地址等参数,可执行对于远程服务器的各类检测,检测命令对于远程服务器的检测有以下两种情形:

(1) 某些类型的检测命令可不经插件,直接检测远程被监控服务器上的监控项,例如通过部署在 Nagios 和 Centreon 监控服务器上的 `check_oracle` 命令直接检测远程 Oracle 数据库服务器上的监听服务是否正常等等。

(2) 检测命令远程调用位于被监控服务器上的各类 Nagios 插件,可返回被检测端正常与否的状态信息,并且附带着一些性能数据。例如部署在 Nagios 和 Centreon 监控服务器上的著名的检测命令——`check_nrpe`。

以上两种检测方式中,第 2 种比较常见,其中最著名的应当是 `check_nrpe` 检测命令。它可以调用位于远程被监控服务器上的 `nrpe` 代理(见 7.2.2 小节介绍),进而驱使位于 `/usr/local/nagios/libexec` 目录下的各类检测插件以获取告警信息和性能数据。

检测命令采集到的各类告警信息可由 Centreon 显示在其 Web 界面上,供管理员实时查看,而 RRDTool 工具可借助后续的一些性能数据描绘出该被监控项的历史趋势图形。

更详细的有关检测命令的内容可参考 11.5 节。

7.5.4 调度进程

按照主动监控模式,Centreon 监控系统中的调度进程(又称调度器 Scheduler 或者调度引擎 Scheduling Engine)按照设定的检测周期定时发起对被监控项的检测,并搜集返回的检测结果及性能信息。除此之外,调度进程还会从事消息通知、定期执行任务等工作。

一般来说,我们对于调度进程的期望和对于检测命令一样,都希望它们对于被检测系统具有最小的侵入性,换句话说,对被检测系统的影响最小。

适用于 Centreon 监控系统的,最广泛和最著名的调度器就是 Nagios,还有根据 Nagios 复制(fork)出来的分支项目,例如 Shinken、Icinga、Centreon Engine 等项目。这些分支项目在保持与 NRPE、NSCA 等 Nagios 模块化组件兼容的前提下,与 Nagios 相比,能提供更多特性、具备更好的性能。

用户使用 Centreon 的 Web 界面就可以管理这些调度器的相关配置,并能够生成合适的配置文件,将配置文件放置在不同的目录中,期间不用对配置文件有任何的编辑、拷贝等人为操作,做到了完全的自动化。

Nagios 调度进程

作为拥有 250000 庞大用户群体的 Nagios,迄今为止是 IT 运维监控领域应用最为广泛的开源产品之一。由于具备对几乎所有设备的兼容性以及数量庞大、用途广泛的各式插件,Nagios 已经成为 Centreon 监控系统的默认调度进程。

Centreon Engine

Centreon Engine 是由 MERETHIS 公司在 2011 年,从 Nagios 的版本 3.2.3 复制(fork)而成的一个分支版本。Centreon Engine 在 Nagios 基础上增加了几个补丁程序,在确保与 Nagios 及其组件(NSCA、NRPE 和 NDOUtils 等)兼容的前提下,显著提升了性能并增加了功能。Centreon Engine 是 Centreon 监控系统的重要组成部分,也是 Centreon 监控系统进

一步演化的重要支柱。Centreon 监控系统的负载均衡、高可用性等突出特性都依赖于 Centreon Engine 的工作表现。更详细的 Centreon Engine 介绍可以在其官方网站 <http://www.centreon.fr/Article-Produits-Centreon-Engine/roadmap> 上找到。

如 1.4 节所述，Nagios 近年来发展缓慢，一直靠少数的开发者维护。很长一段时间内，社区开发者们一直尝试提升 Nagios 的内核功能，为其打各种补丁等等，做了很多工作，但起效不大。这就是为什么近年来涌现出许多 Nagios 的分支版本——Shinken、Icinga、Centreon Engine 等克隆性质的调度引擎的原因。在 2012 年，Centreon Engine 以及 Centreon 监控系统的开发者 MERETHIS 发表了 Nagios 和 Centreon Engine 的比较结果，结论是后者要比前者性能高出 8 到 9 倍，Centreon Engine 大获全胜。截至目前，Centreon Engine 的版本已经是 1.2，且得到了大规模的应用。

7.5.5 其他兼容 Centreon 的调度引擎

其他的一些兼容 Centreon 的调度引擎，例如 Icinga 和 Shinken，几乎和 Centreon Engine 同时出现。由于这两个调度引擎与 Nagios 和 Centreon Engine 仍存在些许不同，因此 Centreon 监控系统仅能够管理其中部分功能。

Icinga 是由一家德国公司于 2009 年开发，同样是基于 GPL V2 协议的开源软件。Icinga 具有自己的 Web 管理界面 Icinga Web，但是同样能够被其竞争对手 Centreon 所管理。

而 Shinken 完全是由法国人 Jean Gabes 使用 Python 脚本语言重新开发的，与 Nagios 兼容的监控调度进程，基于 GNU AGPL license 2 协议授权。Shinken 最初的开发是为了向开发者们证明 Nagios 的核心代码是完全可以升级而变得更加有效，为了达到这一目的，Jean Gabes 使用 Python 语言从头开发了 Nagios，出色地完成任务。事实证明，相较于 Nagios，Shinken 更适合在大规模的 IT 基础设施环境中部署，且运转良好。

7.5.6 代理进程

调度器代理进程（scheduler broker）的任务就是搜集检测命令返回的监控信息，供 Centreon 监控系统的 Web 界面显示。具体来讲，代理进程的功能就是接收所有监控信息，将其存入数据库中，由 Centreon 监控系统以定时轮询的方式（以秒为单位），在 Web 界面上实时显示监控数据。

Centreon 监控系统上最著名的代理进程就是 NDOUtils，但无论如何，它仍然有可能被其竞争对手——Centreon Broker 所取代，后者是 Centreon 公司开发的 NDOUtils 的替代品。

NDOUtils

NDOUtils，有时简称 NDO，是专为 Nagios 开发的数据库访问接口。它的功能是搜集 Nagios 检测到的各类信息，并将其存入到后台 MySQL 数据库中。

NDOUtils 它包括两个程序组件，ndomod 和 ndo2db，如图 7-3 所示。

- ndomod：ndomod 随着 Nagios 的启动而启动，它的作用是将调度进程（Nagios、Centreon Engine、Icinga 或者 Shinken）搜集的各类监控数据和性能数据拷贝到文件中，并将这些文件发往一个 Unix 网络套接字（Unix socket）中。

- ndo2db: ndo2db 是运行在 Nagios 或者 Centreon 监控服务器上的一个后台进程。它的主要任务是从 ndomod 使用的 Unix 网络套接字中获取后者发送的各类监控和性能数据，将其存入后台的 centreon_status 数据库中，供 Centreon 监控系统实时显示在 Web 界面上。

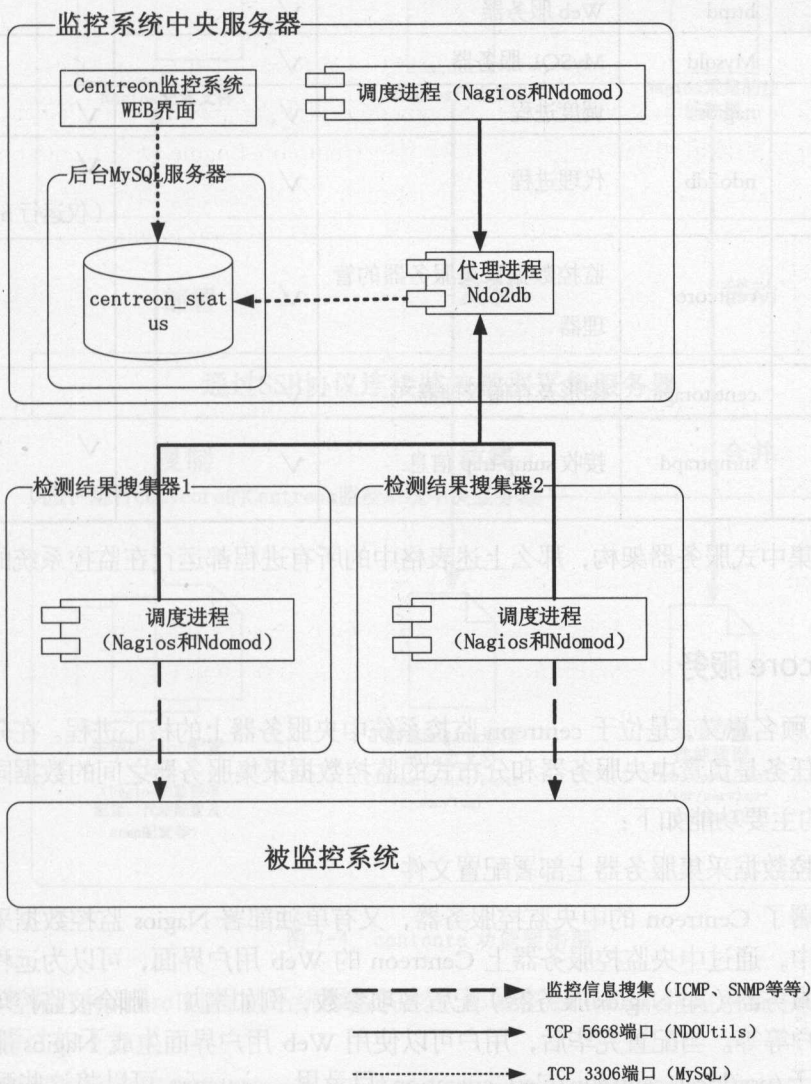


图 7-3 NDOUtils 架构图

7.6 后台服务和定时任务

以下表格列出了 Centreon 监控系统正常运行所需要的一些后台服务和定时任务。在 Nagios 分布式服务器架构中，这些后台服务和定时任务在监控系统的中央监控服务器上，以及部署在不同位置的监控数据采集服务器（同样部署了 nagios 调度进程）上都有运行，如表 7-3 所示。

表 7-3 Centreon 监控系统后台服务及定时任务表

操作系统后台服务	进 程	描 述	中心监控服务器	监控数据采集服务器
httpd	httpd	Web 服务器	√	
mysql-server	Mysqld	MySQL 服务器	√	
nagios	nagios	调度进程	√	√
ndoutils-ndo2db ndoutils-ndomod	ndo2db	代理进程	√	√ (仅运行 ndomod)
centreon	centcore	监控数据采集服务器的管理器	√	
centreon	centstorage	图形及存储管理器	√	
net-snmp	snmptrapd	接收 snmp trap 信息	√	√ (可选)

如果采用集中式服务器架构，那么上述表格中的所有进程都运行在监控系统的中央监控服务器上。

7.6.1 centcore 服务

centcore，顾名思义，是位于 centreon 监控系统中央服务器上的核心进程。在分布式服务器架构中，其任务是负责中央服务器和分布式的监控数据采集服务器之间的数据同步工作。

centcore 的主要功能如下：

(1) 在监控数据采集服务器上部署配置文件

在既有部署了 Centreon 的中央监控服务器，又有单独部署 Nagios 监控数据采集服务器的分布式环境中，通过中央监控服务器上 Centreon 的 Web 用户界面，可以为远程分布式的监控数据采集服务器（即 Nagios 服务器）配置各项参数，例如增加、删除被监控项、管理检测项、管理用户等等。当配置完毕后，用户可以使用 Web 用户界面生成 Nagios 服务器的配置文件——位于 /usr/local/centreon/filesGeneration/ 目录里。centcore 可以将这些配置文件以 SSH 的方式传输到相应的监控数据采集服务器上。

(2) 管理监控数据采集服务器上的调度进程

用户可以在 Centreon 监控系统的 Web 用户界面上对远程监控数据采集服务器上的调度进程，例如 Nagios 进程，进行启动（start）、停止（stop）和重启（restart）等操作。

集中搜集并存储监控数据采集服务器采集到的日志信息和性能信息。

一般来说，监控数据采集服务器（即 Nagios 服务器）的日志信息路径如下 /usr/local/nagios/var/nagios.log 下，而被监控项的性能数据信息位于 /usr/local/nagios/var/service-perfdata 目录里。

centcore 功能架构图如图 7-4 所示。

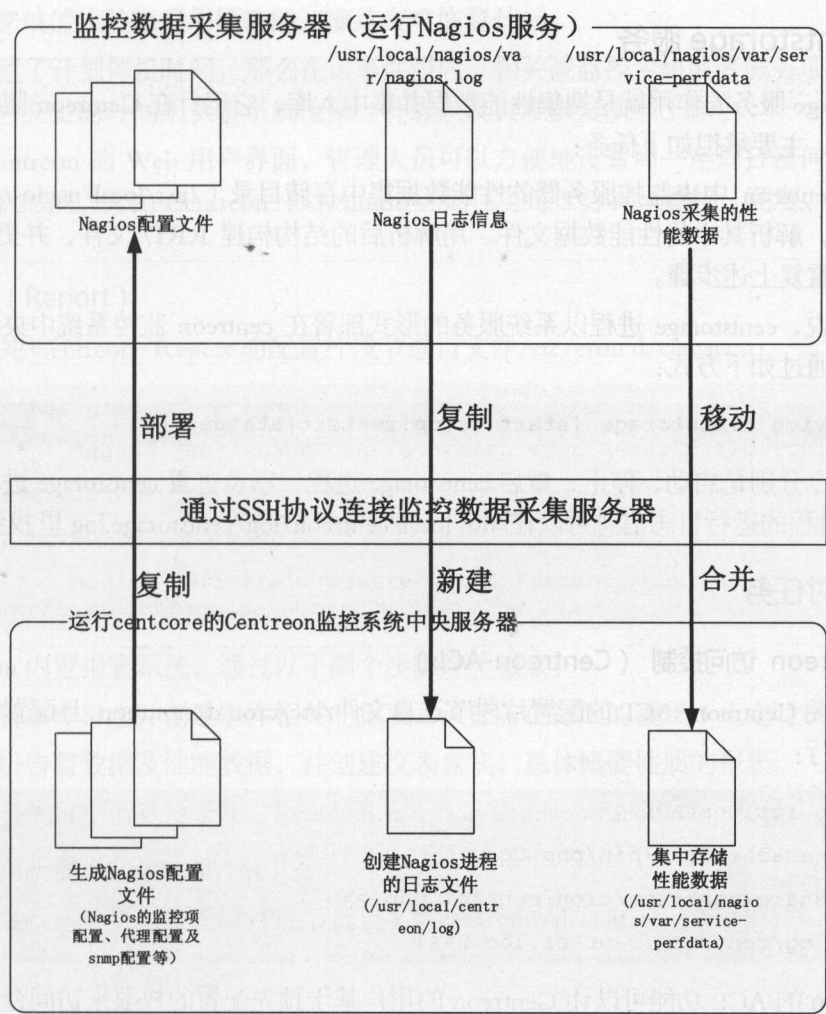


图 7-4 centcore 功能架构图

一般来说，centcore 进程以系统服务的形式部署在 centreon 监控系统中央服务器上，其管理可以通过如下方式：

```
# service centcore {start |stop|restart|status}
```

以上命令分别是启动、停止、重启 centcore 进程，以及查看 centcore 进程的状态。
centcore 进程的运行日志信息可以在 /usr/local/centreon/log/centcore.log 里找到。

注意：centcore 组件只有在 Centreon 监控系统采用分布式服务器部署架构时才用到。

关于 SNMP trap 监控项的配置也是通过 centcore 来进行的，所产生的相关配置位于中央服务器的 /etc/snmp/centreon_traps。

为了使 Centreon 监控系统中央服务器能够与位于远程的分布式 Nagios 服务器之间进行 SSH 数据交换，两者之间事先必须通过先交换密钥来建立连接及信任关系。

7.6.2 centstorage 服务

centstorage 服务的作用就是搜集性能数据并集中入库，它运行在 Centreon 监控系统的中央服务器上，主要承担如下任务：

访问 Centreon 中央监控服务器的性能数据集中存储目录（/usr/local/nagios/var/service-perfdata）中，解析其中的性能数据文件。用解析后的结构构建 RRD 文件，并更新到 RRD 数据库中。重复上述步骤。

一般来说，centstorage 进程以系统服务的形式部署在 centreon 监控系统中央服务器上，其管理可以通过如下方式：

```
# service centstorage {start |stop|restart|status}
```

以上命令分别是启动、停止、重启 centstorage 进程，以及查看 centstorage 进程的状态。centstorage 进程的运行日志信息可以在 /usr/local/centreon/log/centstorage.log 里找到。

7.6.3 定时任务

1. Centreon 访问控制（Centreon-ACL）

以下有关 Centreon-ACL 的配置片段节选自文件 /etc/cron.d/centreon，且配置命令仅为一行，不能换行：

```
# Cron for CentACL
***** apache /usr/bin/php Cq
/usr/share/centreon/cron/centAcl.php >>
/var/log/centreon/centAcl.log 2>&1
```

Centreon 的 ACL 功能可以让 Centreon 的用户基于预先配置的权限来访问合适的菜单项或者页面。换句话说，基于预先设定的用户组（User Group），Centreon 的某些用户可以被限制为只能访问具备权限的页面，而未被分配访问权限的页面则无法访问。

Centreon-ACL 计划任务的功能是，基于用户在 Centreon 的 Web 界面中创建好的 ACL 访问控制细则，产生相应的全局权限配置文件。该计划任务可以每分钟调度一次，也可以根据 Centreon Web 用户界面的设置周期而调度。

2. 计划停机时间（Centreon-Downtime）

以下有关 Centreon-Downtime 的配置片段节选自文件 /etc/cron.d/centreon：

```
# Cron for Centreon-Downtime
*/5 * * * * apache /usr/bin/php -q
/usr/share/centreon/cron/downtimeManager.php
>> /var/log/centreon/downtimeManager.log 2>&1
```

在 Centreon 监控系统中存在“计划停机时间（Downtime）”的概念，在大型 IT 环境中，处于服务可靠性和系统稳定性的考虑，有必要通过设置计划停机时间，如果能充分利用计划

停机窗口，您就能检验高可用性功能、演练灾难恢复计划。

如果设置了计划停机时间，那么在该事件段内，相关被监控主机或者服务项将不会被监控，用户会有充足的时间来从事系统检修、升级以及灾难恢复演练工作。

通过 Centreon 的 Web 用户界面，管理人员可以方便地设置周一至周日任何一个时间段内的计划停机时间，并由 Centreon-Downtime 计划任务每 5 分钟来扫描、调度、实施这些停机计划。

3. 报表 (Report)

以下有关 Centreon-Report 的配置片段节选自文件/etc/cron.d/centreon:

```
# Cron to build State events
0 3 * * * nagios /usr/share/centreon/cron/eventReportBuilder.pl -
1 >> /var/log/centreon/eventReportBuilder.log 2>&1

# Cron to build reporting
0 6 * * * nagios /usr/share/centreon/cron/dashboardBuilder.pl -l i >>
/var/log/centreon/dashboardBuilder.log 2>&1
```

Centreon 内置报表系统，通过以下两个步骤产生报表：

- (1) 解析事件日志并创建事件相关的报表，该任务在每天的凌晨 3 点定时执行。
- (2) 解析告警数据及性能数据，并创建仪表盘式，总体概要性质的报表。

以上任务的日志信息位于/usr/local/centreon/log/dashboardBuilder.log 文件中。

4. 日志分析器 (Log analyser)

以下有关日志分析器的配置片段节选自文件/etc/cron.d/centreon:

```
# Cron for log parsor
***** nagios /usr/share/centreon/bin/logAnalyser >>
/var/log/centreon/logAnalyser.log 2>&1
```

日志分析器 Log analyser 每分钟记录一次 Nagios 调度进程的监控项告警日志，通过 centstorage 服务，将搜集到的告警日志存储到后台数据库中。通过 Centreon 监控系统 Web 用户界面的 Event Logs 菜单，可以看到这些搜集到的告警日志信息。

日志分析器 Log analyser 的工作日志文件全路径是/usr/local/centreon/log/logAnalyser.log。

■ 性能搜集器 (Performance collector)

以下有关性能搜集器的配置片段节选自文件/etc/cron.d/centreon:

```
# Cron for tracing Nagios Poller Performances
*/5 * * * * nagios /usr/share/centreon/bin/nagiosPerfTrace >>
/var/log/centreon/nagiosPerfTrace.log 2>&1
```

如上述配置片段，nagiosPerfTrace 进程每 5 分钟调度一次 nagiosstats 进程，以 SSH 协议访问 Centreon 监控系统中央服务器，搜集后台 Mysql 数据库中的监控性能数据，并将这些数据

构建成为 RRD 图形文件，存放在 /var/lib/centreon/nagios-perf 目录中。其工作日志是文件 /usr/local/centreon/log/nagios-PerfTrace.log。

通过 Centreon 监控系统的 Web 用户界面可以查看到 nagios 的性能统计信息。

■ 后台清除定时任务 (Purge)

同样是查看定时任务配置文件 /etc/cron.d/centreon:

```
# Cron for databin and logs purge
0 2 * * * nagios /usr/local/centreon/cron/centreonPurge.sh >>
/usr/local/centreon/log/centreon-purge.log 2>&1
```

再查看上述脚本中提及的 centreonPurge.sh 文件:

```
sudo /etc/init.d/centstorage stop
# Wait a little
sleep 10
/usr/local/centreon/cron/purgeCentstorage >>
/usr/local/centreon/log/centreon-purge.log 2>&1
# Wait a little
sleep 10
/usr/local/centreon/cron/purgeLogs >>
/usr/local/centreon/log/centreon-purge.log 2>&1
# Wait a little
sleep 10
sudo /etc/init.d/centstorage start
exit 0
```

脚本 centreonPurge.sh 的功能就是基于操作系统 cron 定时任务，在相应日期的凌晨 2 点清除后台 Mysql 数据库中，以及 /var/lib/centreon 目录下的，由 centstorage 服务进程存储的性能数据信息和告警日志。该脚本的相关日志输出位于文件 /usr/local/centreon/log/centreon-purge.log 中。

7.7 系统架构——简洁及分布式

Centreon 的分布式架构中，信息流程清晰明确，各个组件各司其职，分工明确，共同执行着如图 7-5 所示的信息流程：

ndo2db 组件负责采集实时监控信息和性能数据，而 Centreon 的 Web 界面负责展示这些数据。

调度进程 Nagios 生产并展示 RRD 数据库文件，并组装 Nagios 的配置文件。

Centreon 的 Web 用户接口展示监控代理搜集到的信息系统实时告警信息。

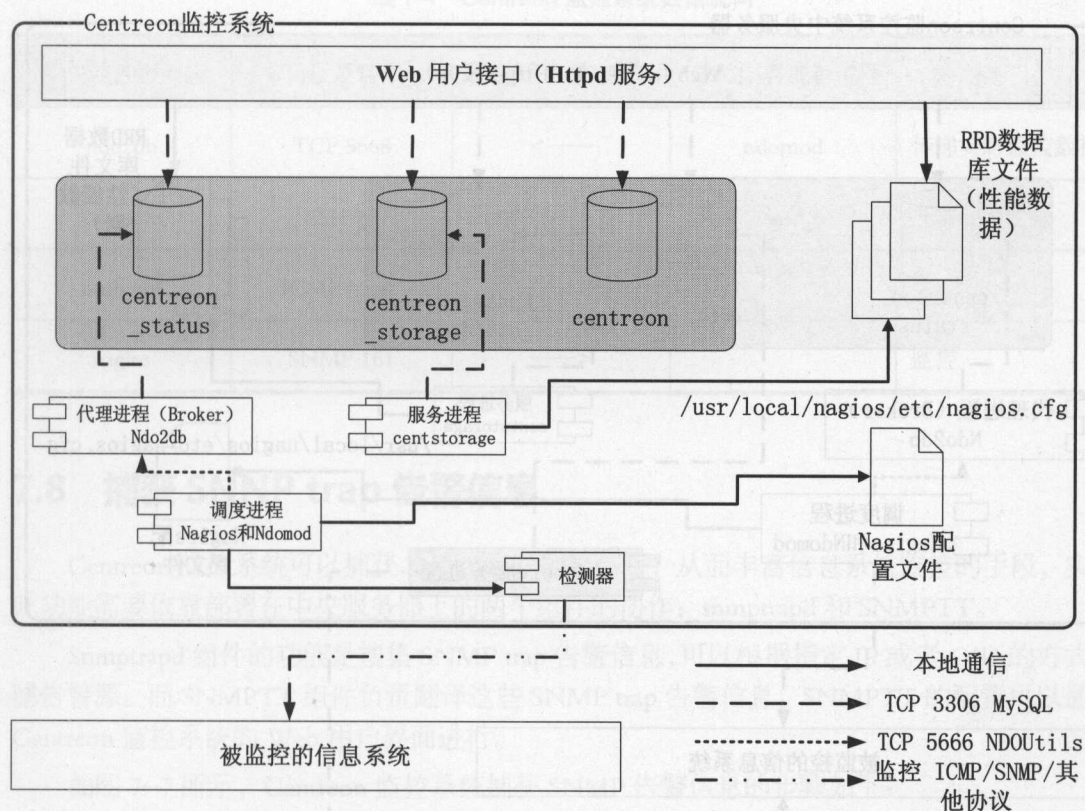


图 7-5 Centreon 监控系统架构图

所谓分布式架构，指的是 Centreon 监控系统采用的中央服务器和分布式监控数据采集服务器之间协作运行的架构。分布式数据采集服务器采集所辖网络范围内的信息系统的各类告警信息和性能数据，并将结果上报给中央监控服务器。相较于集中式服务器的一元架构来说，分布式架构能够降低网络负载和监控系统负载，提升监控系统的性能。

在分布式架构中，采集服务器可独立运作。一旦采集服务器与中央监控服务器之间的网络链路故障，采集服务器仍然会保持对于信息系统的监控，当网络恢复后，会将故障期间的性能数据同步至中央监控服务器。在网络安全方面，分布式的架构有利于服务器与服务器之间的开放端口降低到最少，可以最大限度地避免出现黑客攻击等问题。

作为 Centreon 监控系统分布式架构的核心，起到大脑和灵魂作用的就是部署在中央服务器上的 centcore 服务进程。centcore 服务进程与分布式监控数据采集服务器之间的通信方式有两种方向，第一种方向是 centcore 进程下发配置文件到采集服务器上，相反方向是从采集服务器那里搜集监控信息和性能数据。centcore 服务进程的另一项重要工作就是，将搜集到的监控信息和性能数据发送给 centstorage 服务进程，处理并存储到后台 MySQL 数据库中。

Centreon 架构如图 7-6 所示。

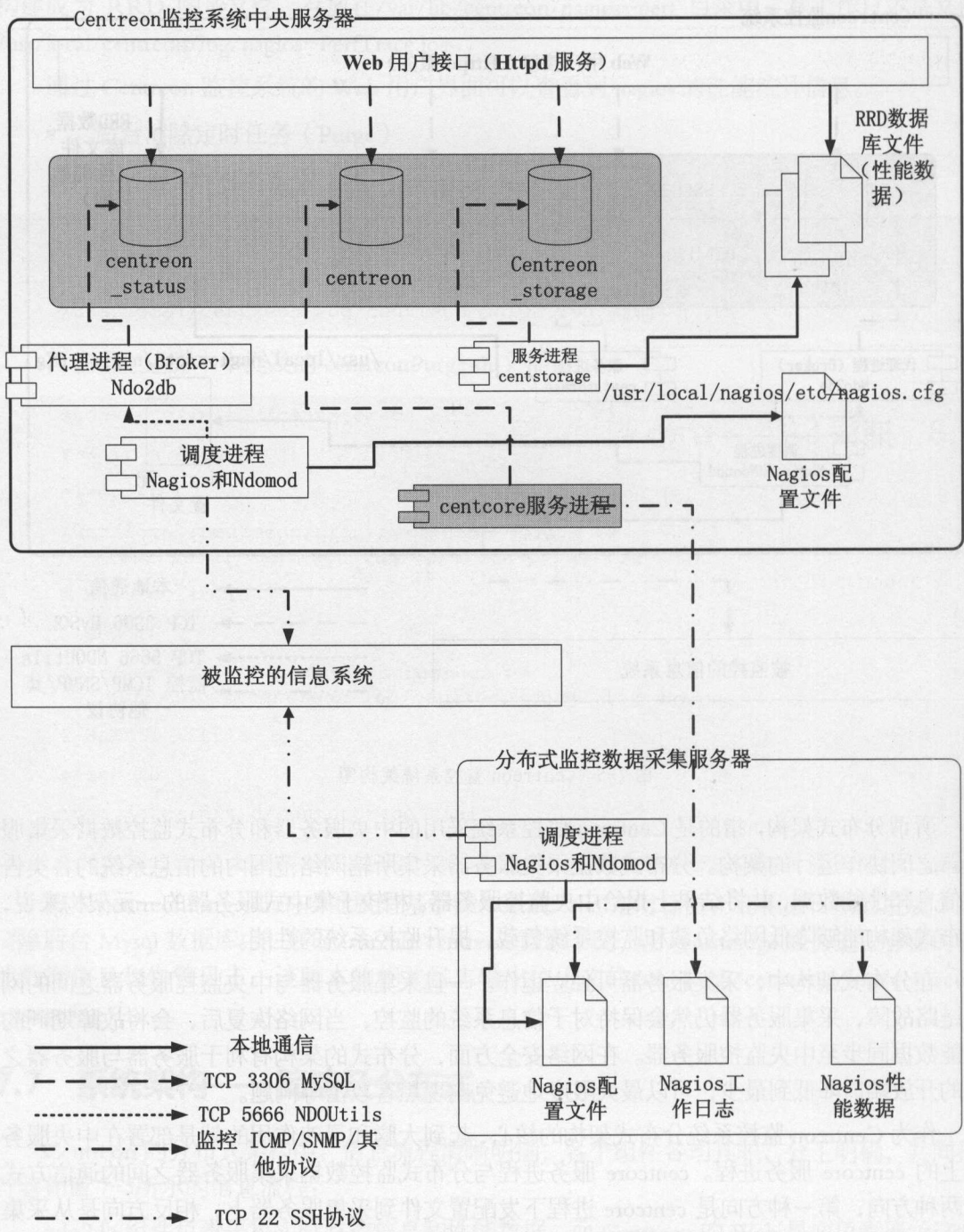


图 7-6 Centreon 分布式架构图

表 7-4 描述了中央监控服务器与分布式监控数据采集服务器之间的数据流。

表 7-4 Centreon 监控系统数据流向

中央监控服务器	协议及端口	数据传送方向	依赖服务或者进程	描 述
ndo2db	TCP 5668	<—	ndomod	传递实时监控数据
centcore	SSH 22	—>	sshd	部署配置文件
centcore	ICMP (ping)	—>		监控
nagios	SNMP 161	—>		监控

7.8 捕获 SNNP trap 告警信息

Centreon 监控系统可以捕获 SNMP trap 告警信息,从而丰富信息系统监控的手段,实现此功能需要依靠部署在中央服务器上的两个组件的协作: snmptrapd 和 SNMPTT。

Snmpttrapd 组件的功能是搜集 SNMP trap 告警信息,可以根据指定 IP 或者 OID 的方式过滤告警源。而 SNMPTT 组件负责翻译这些 SNMP trap 告警信息。SNMPTT 的配置可以通过 Centreon 监控系统的 Web 用户界面进行。

如图 7-7 所示, Centreon 监控系统捕获 SNMP 告警信息的步骤如下:

- (1) snmptrapd 组件监听 162 端口,捕获 SNMP Trap 告警信息。
- (2) SNMPTT 负责翻译这些告警信息。
- (3) 由 centTrapHandler-2.x 组件根据 Centreon 事先的配置,负责将这些 SNMP trap 告警信息传递给被动监控模式下的 Nagios 监控调度进程。

从部署方式来看,无论是采用集中式服务器架构还是分布式架构,捕获 SNMP trap 告警信息的相关组件都位于 Centreon 监控系统中央监控服务器上。

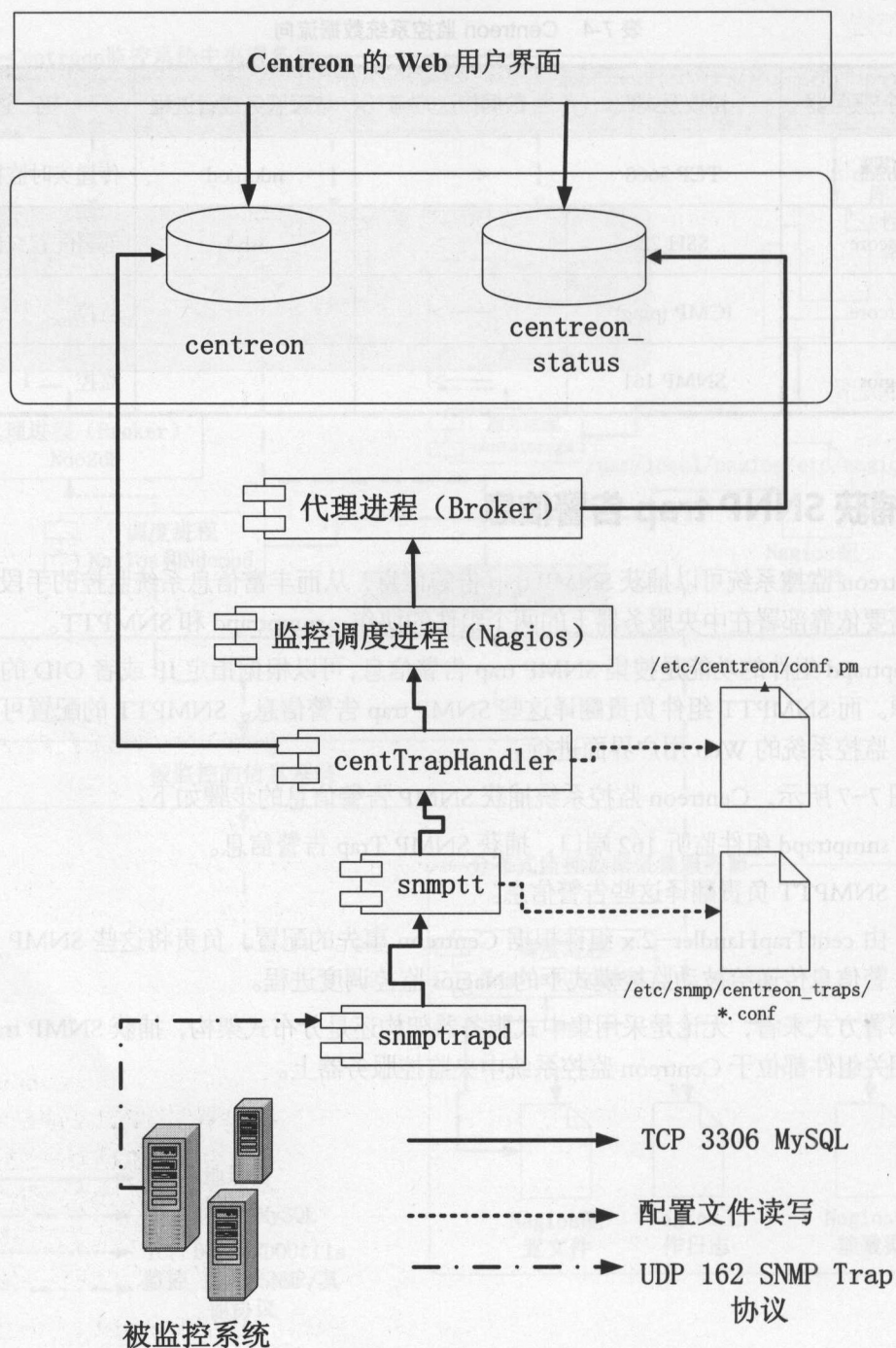


图 7-7 Centreon 捕获 SNMP trap 告警信息功能架构图

第 8 章

安装 Centreon

Centreon 是开源的 IT 监控软件，由法国人于 2003 年开发，最初名为 Oreon，并于 2005 年正式更名为 Centreon。

Centreon 作为 Nagios 的分布式监控管理平台，功能强大，在 IT 监控方面地位强势，它的底层使用 Nagios 监控软件，Nagios 通过 ndoutils 模块将监控数据写入数据库，Centreon 读取该数据并即时的展现监控信息，通过 Centreon 可以简单地管理和配置所有 Nagios，因此，完全可以使用 Centreon 轻松搭建企业级分布式 IT 基础运维监控系统。

——来自好搜百科

8.1 安装前提

Centreon 监控系统所在服务器的硬件配置依赖于监控项的数量，一般来说，根据表 8-1 所示，用户可以选择适合自身的服务器硬件配置：

表 8-1 Centreon 监控系统服务器配置参照表

监控项数量	部署方式	服务器处理器数量	中央监控服务器内存
< 500	1 中央服务器	1 × 2.5 GHz	1 GB
500 - 1000	1 中央服务器	2 × 2.5 GHz	2 GB
1000 - 2000	1 中央服务器	4 × 2.5 GHz	4 GB
2000 - 5000	1 中央服务器+1 监控数据采集服务器	2 × 4 × 2.5 GHz	2 × 8 GB
5000 - 10000	1 中央服务器+2 监控数据采集服务器	3 × 4 × 2.5 GHz	4 × 8 GB

接下来是安装并部署 Centreon 和 Nagios 监控系统所需的磁盘空间预估。在 Centreon 的所有组件中，MySQL 数据库和 RRD 图形数据库文件是磁盘空间消耗大户，通常位于中央监控服务器上，而分布式的监控数据采集服务器占据的磁盘空间并不大。

对于系统管理员最关心的数据有效周期而言，一切都可以通过 Centreon 的 Web 用户界面来进行配置：数据保留周期可以任意指定，过期数据可以由 cron 进程调度相应脚本自动清除等等。基于以上配置，中央监控服务器上所需的 MySQL 数据库及 RRD 图形文件存储所需的空间就可以进行估算了，一般来说，这种估算需要借助于下列 3 项参数：

- 监控项的数量；
- 两次检查之间的平均时间；
- 检测数据保留时间。

1. MySQL 数据库占用空间估算

Nagios 每执行一次检测，后台 MySQL 数据库中就会多几条相关的记录。随着检测次数的增多，后台数据库的相关记录会逐渐增多，根据检测数据保留期限，数据库的占用空间很快会达到一个定值。

表 8-2 是在 Centreon 实际部署过程中观测到的一些常规参数：

表 8-2 Centreon 实际设置参数

单个监控项的性能检测消息个数	2
单个性能检测消息的平均大小	54 字节（Byte）
单个 RRD 图形数据库文件的平均大小	默认保存 6 个月，平均大小为 845KB

对于监控数据（含告警的字符串信息和性能数据）占用空间的总体大小，我们可以用检

测数据量乘以每条数据的平均大小来计算。

其中的检测数据量可以用检测数据保留时间段内的检测次数乘以所部署监控项的数量以及每个监控项的平均检测次数。

计算方式如下：

$$e = \frac{60}{a} \times 24 \times 30 \times r \times s \times 2 \times 54 \times 10^{-9} = \frac{4.7 \times r \times s}{1000 \times d}$$

- e 代表所占用的磁盘空间，单位是 GB，即千兆字节。
- d 代表两次检查之间间隔的分钟数。
- r 代表检测数据保留的月数。
- s 代表监控项的数量。

例如，对于部署了 1000 个监控项的 Centreon 监控系统，检测数据保留期限为 12 个月。每个监控项的平均检测间隔为 5 分钟，那么该监控系统的 MySQL 数据库所需空间计算如下：

$$e = 4.7 \times \frac{12 \times 1000}{1000 \times 5} = 11.2 \text{ GB}$$

2. RRD 图形数据库文件占用空间估算

在 Centreon 监控系统中，每个 RRD 图形文件代表一个监控项的性能数据图形。RRD 文件是固定的，在创建之初即拥有定义的最大尺寸。RRD 图形文件占用的空间可以用检测数据保留期限内的性能数据图形数量（即在 Centreon 中被配置为能够展示性能图形的监控项）乘以 RRD 文件的尺寸来计算。

计算方式如下：

$$e = s \times (2 + 1) \times \frac{84500 \times r}{6 \times d} \times 10^{-9} = \frac{4225 \times r \times s}{d \times 10^6}$$

- e 代表 RRD 图形文件所占用的磁盘空间，单位是 GB，即千兆字节。
- d 代表两次检查之间间隔的分钟数。
- r 代表检测数据保留的月数。
- s 代表监控项的数量。

例如，1000 个配有性能数据图形的监控项，平均检测间隔时间为 5 分钟，图形文件保留期限为 12 个月，那么其 RRD 图形数据库文件总体大小计算如下：

$$e = \frac{4225 \times 12 \times 1000}{5 \times 10^6} = 1 \text{ GB}$$

上述计算结果，再加上操作系统、Centreon 监控系统、各类第 3 方应用软件和组件所需的存储空间，还需要增加大约 5GB 的磁盘空间。安全起见，建议要在上述计算基础上再增加一倍的存储空间。下述公式给出了 Centreon 监控系统中央服务器的总体磁盘容量需求：

$$e = \frac{7.55 \times r \times s}{1000 \times d} + 5$$

根据上述计算，1000 个监控项，平均检测间隔为 5 分钟，检测数据保留期限为 12 个月，

那么总体的中央服务器磁盘容量需求为 25GB。

在 Centreon 监控系统中，仅当被监控主机及服务项的状态发生变化时，才记录日志，因此日志文件并不是很庞大。但需要注意的是，如果启用了 Nagios 等代理进程的详细记录模式，由于需要记录的项目增多，那么日志文件仍有可能变得较大。

注意：除了监控重要的 IT 系统之外，对监控系统服务器自身的监控，包括文件系统、负载、换页空间、内存，以及硬件监控也是非常有必要的。

8.2 安装 Centreon 监控系统中央服务器

8.2.1 系统软件需求

要想正确安装并运行 Centreon 监控系统，首先要检查下列 Linux 操作系统软件包及 PHP 组件的安装情况。

- rrdtool

可用操作系统自带的“添加/删除软件”工具安装该软件包，如图 8-1 所示。

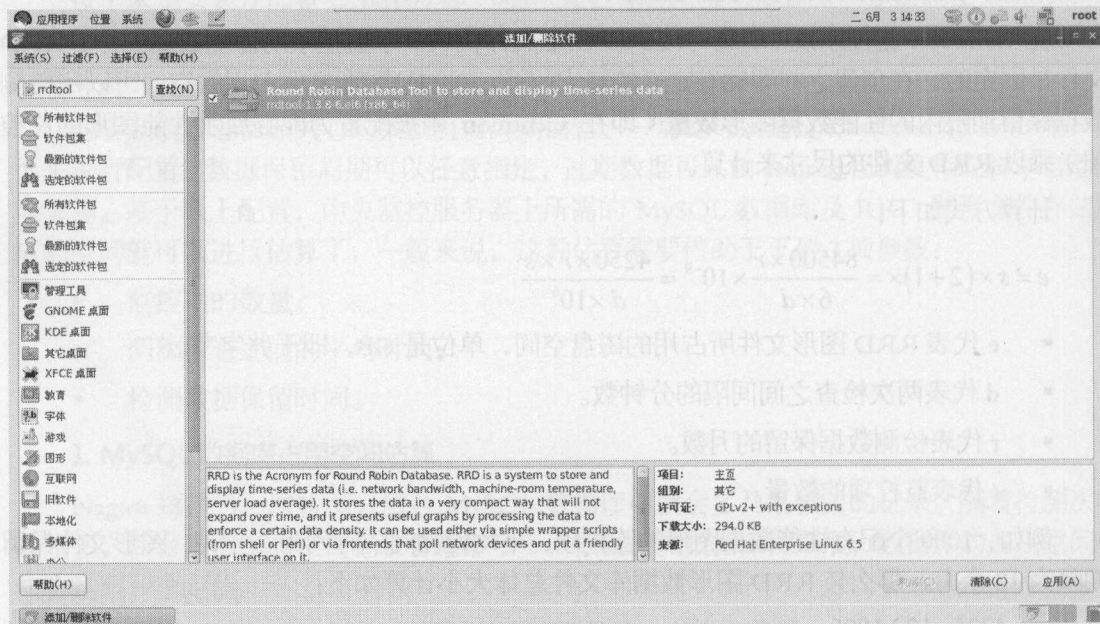


图 8-1 安装 rrdtools

- perl-snmp

由于很多监控插件基于 Perl 语言编写，且采用了 SNMP 方式搜集被监控端的各类信息，因此有必要安装 perl-snmp 组件，为 Perl 提供 SNMP 检测方面的支持。如图 8-2 所示。

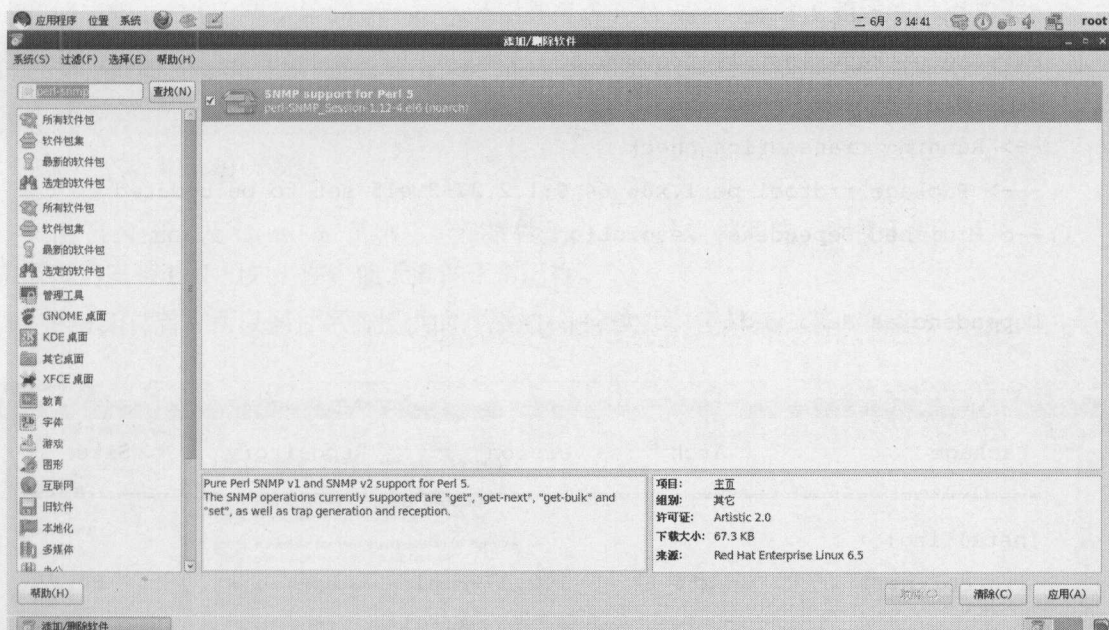


图 8-2 安装 perl-snmpp 软件包

■ perl-rrdtool

为了使 perl 与 rrdtool 能够协同工作,必须为 perl 语言运行环境添加 rrdtool 的组件支持可以使用下列命令检测系统中是否存在 perl-rrdtool 组件。

```
[root@monitor perl5]# perl -MRRDs -le 'print q(ok!)'
Can't locate RRDs.pm in @INC (@INC contains: /usr/local/lib64/perl5
/usr/local/share/perl5 /usr/lib64/perl5/vendor_perl
/usr/share/perl5/vendor_perl /usr/lib64/perl5 /usr/share/perl5 .).
BEGIN failed--compilation aborted.
```

上述输出表明系统未安装 rrdtool-perl 组件,那么在系统联网的前提下,我们可以使用 yum 命令安装该组件。

```
# yum install rrdtool-perl
Loading "fastestmirror" plugin
Loading mirror speeds from cached hostfile
* epel: mirrors.ircam.fr
* base: mirrors.service.softlayer.com
* updates: mirrors.service.softlayer.com
* addons: mirrors.service.softlayer.com
* extras: mirrors.service.softlayer.com
Excluding Packages in global exclude list
Finished
```

```

Setting up Install Process
Parsing package install arguments
Resolving Dependencies
--> Running transaction check
---> Package rrdtool-perl.x86_64 0:1.2.27-3.el5 set to be updated
--> Finished Dependency Resolution

Dependencies Resolved

=====

Package                Arch      Version      Repository    Size
=====
Installing:
rrdtool-perl            x86_64    1.2.27-3.el5  epel          34 k

Transaction Summary

=====
Install      1 Package(s)
Update      0 Package(s)
Remove      0 Package(s)

Total download size: 34 k
Downloading Packages:
(1/1): rrdtool-perl-1.2.2 100% |=====| 34 kB 00:00
Running rpm_check_debug
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction

  Installing: rrdtool-perl ##### [1/1]

Installed: rrdtool-perl.x86_64 0:1.2.27-3.el5
Complete!
Test shows it should now be available.

```


使用如下命令，再次检测 `rrdtool-perl` 组件是否安装正常，正确安装后的输出如下。

```
$ perl -MRRDs -le 'print q(ok!)'
ok!
```

■ Perl 相关组件支持

由于 Nagios 及 Centreon 的很多检测插件是由 Perl 语言编写的，为便于插件正常运行，需要提前安装 Perl 相关组件，便于插件正常运行。

Perl 组件首先可以通过系统提供的“添加/删除软件”工具来安装，如图 8-3 所示。

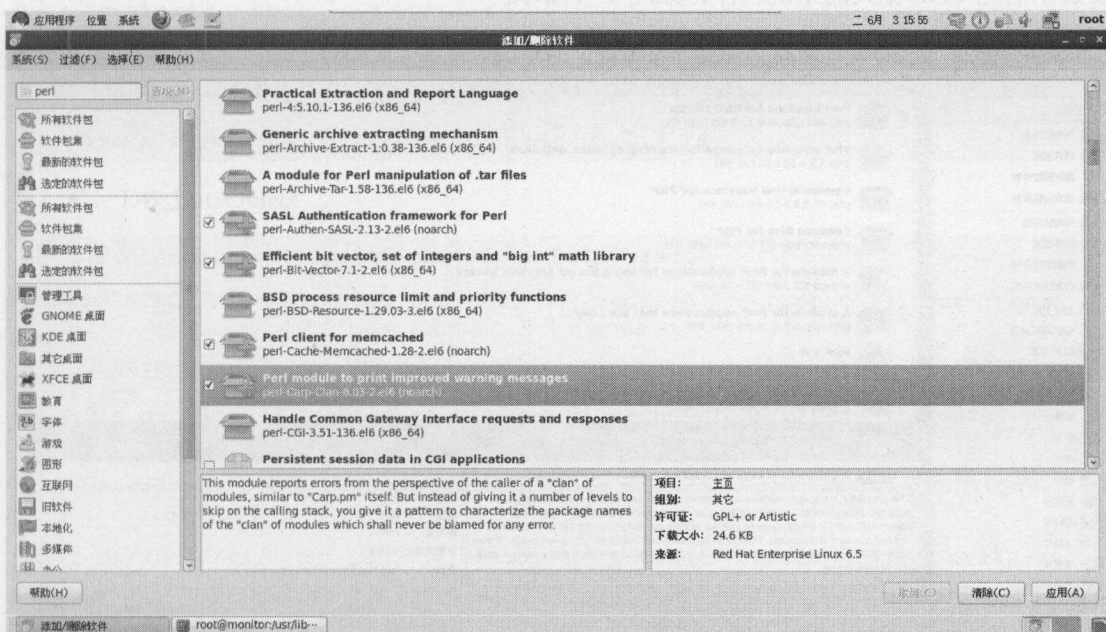


图 8-3 Perl 及其组件安装

除了系统提供的工具之外，还可以通过著名的 CPAN 网站来下载各类 Perl 扩展应用包及插件，丰富 Perl 的功能。CPAN (Comprehensive Perl Archive Network) 中译为“Perl 程序库”，它包含了极多用 Perl 写成的软件和其文件，其作用是让 Perl 的开发者和使用者容易从 CPAN 下载、安装、更新及管理其他在 CPAN 上的 Perl 程序，以扩展 Perl 的功能。

■ PHP 及相关扩展应用库支持

PHP 是一个非常优秀的脚本语言，简洁、高效，随着 4.0 的发布，越来越多的人使用它来进行动态网站的开发，可以说，PHP 已经成为最优秀的 INTERNET 开发语言之一，尤其对于那些需要能够快速、高效地开发中小规模的商业应用的网站开发人员，PHP 是其首选的语言。但是随着 PHP 的应用的不断增多，对于这些应用缺乏统一的标准和有效的管理，因此，PHP 社区很难像 PERL 社区的人们那样方便的共享彼此的代码和应用，因为 PHP 缺乏像 CPAN 那样的统一的代码库来分类管理应用的代码模块（熟悉 PERL 的人都知道，CPAN 是一个巨大的 PERL 的扩展模块仓库，编写的应用模块可以放在 CPAN 下面的适当的分类目录

下面，其他的人可以很方便地复用，当然，你编写应用模块时候也需要遵守其中的准则。）

为此，PEAR 就应运而生了，并且从 4.04 开始，随着 PHP 核心一起被分发。PEAR 是 PHP 扩展与应用库（the PHP Extension and Application Repository）的缩写。它是 PHP 扩展及应用的一个代码仓库，简单地说，PEAR 就是 PHP 的 CPAN。

Centreon 监控系统的绝大多数代码是由 PHP 编写的，其正常运转需要 PHP 及其扩展应用库 PEAR 相关组件的支持。下面我们就逐步检查并安装 PHP 及其 PEAR 组件，为后续的 Centreon 监控系统安装做好准备。

使用“添加/删除软件”安装 PHP 相关系统软件，可以使用系统提供的软件安装工具来安装所有与 PHP 相关的系统软件，如图 8-4 所示。

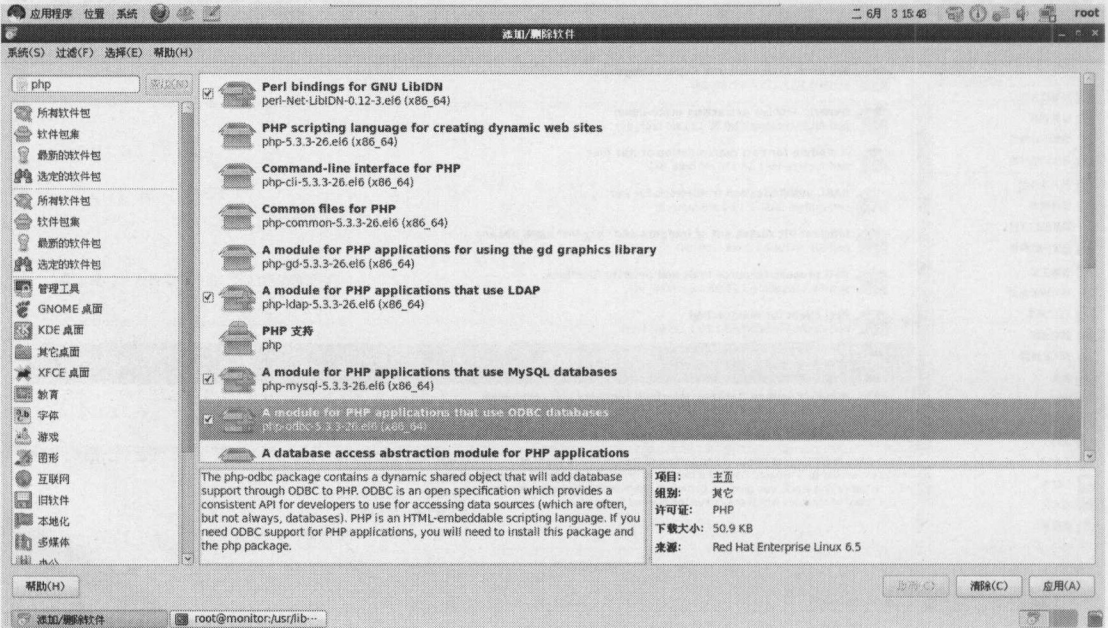


图 8-4 PHP 系统软件安装

在命令行执行下列语句。

```
# pear list
Installed packages, channel pear.php.net:
=====
Package          Version State
Archive_Tar      1.3.7   stable
Console_Getopt   1.2.3   stable
PEAR              1.9.4   stable
Structures_Graph 1.0.4   stable
XML_RPC          1.5.4   stable
XML_Util         1.2.1   stable
```

使用 pear list 命令查看系统 PHP 扩展与应用库时发现缺失了很多组件，因此有必要进一

步安装相关组件。如果系统联网的话,可以在线安装,否则需要到 <http://pear.php.net/index.php> 网站上下载后上传到服务器上再手工安装。

以下是部署和运行 Centreon 监控系统所需要的 PEAR 包:

- DB

```
[root@monitor ~]# pear install DB
WARNING: "pear/DB" is deprecated in favor of "pear/MDB2"
WARNING: channel "pear.php.net" has updated its protocols, use "pear
channel-update pear.php.net" to update
downloading DB-1.7.14.tgz ...
Starting to download DB-1.7.14.tgz (133,103 bytes)
.....done: 133,103 bytes
install ok: channel://pear.php.net/DB-1.7.14
```

- DB_DataObject

```
[root@monitor ~]# pear install DB_DataObject
WARNING: channel "pear.php.net" has updated its protocols, use "pear
channel-update pear.php.net" to update
WARNING: "pear/DB" is deprecated in favor of "pear/MDB2"
Did not download optional dependencies: pear/MDB2, pear/Validate, use
--alldeps to download automatically
pear/DB_DataObject can optionally use package "pear/MDB2" (version >=
2.0.0RC1)
pear/DB_DataObject can optionally use package "pear/Validate" (version >=
0.1.1)
downloading DB_DataObject-1.11.3.tgz ...
Starting to download DB_DataObject-1.11.3.tgz (81,873 bytes)
.....done: 81,873 bytes
downloading Date-1.4.7.tgz ...
Starting to download Date-1.4.7.tgz (55,754 bytes)
...done: 55,754 bytes
install ok: channel://pear.php.net/Date-1.4.7
install ok: channel://pear.php.net/DB_DataObject-1.11.3
```

其余需要安装的扩展与应用库名称分别为 DB_DataObject_FormBuilder、MDB2、Date、HTML_Common、HTML_QuickForm、HTML_QuickForm_advmultiselect、HTML_Table、Archive_Tar、Auth_SASL、Console_Getopt、Image_GraphViz、Mail、Net_DIME、ET-IDNA_0.8.1、Net_SMTP、Net_Socket、Net_Traceroute-0.21.3、Net_Ping、Numbers_Words-0.16.4、PHPUnit、PHP_Compat、Validate-0.8.5、XML_RPC、SOAP-0.13.0 以及 Log。

以上应用库都可以用“pear install 应用库名”的命令来安装,如果某些应用库需要依赖其他库,只需在命令行后加上“—alldeps”参数即可,例如“pear install 应用库名——alldeps”。

所有的 PEAR 应用库都可以在网站 <http://pear.php.net/index.php> 上下载。待以上 PHP 组件安装完毕后，可执行 `pear list` 命令检查输出，符合下列输出即可。

```
[root@monitor]# pear list

Installed packages, channel pear.php.net:

=====

Package                                Version State
Archive_Tar                            1.3.11  stable
Auth_SASL                              1.0.6   stable
Console_Getopt                          1.3.1   stable
DB                                        1.7.14  stable
DB_DataObject                           1.11.3  stable
DB_DataObject_FormBuilder               1.0.2   stable
Date                                     1.4.7   stable
HTML_Common                             1.2.5   stable
HTML_QuickForm                          3.2.13  stable
HTML_QuickForm_advmultiselect           1.5.1   stable
HTML_Table                              1.8.3   stable
HTTP_Request                            1.4.4   stable
Image_GraphViz                          1.3.0   stable
Log                                       1.12.7  stable
MDB2                                     2.4.1   stable
Mail                                     1.2.0   stable
Math_BigInteger                         1.0.2   stable
Net_DIME                                1.0.2   stable
Net_IDNA                                0.8.1   beta
Net_Ping                                 2.4.5   stable
Net_SMTP                                1.6.2   stable
Net_Socket                              1.0.14  stable
Net_Traceroute                          0.21.3  alpha
```


Net_URL	1.0.15	stable
Numbers_Words	0.16.4	beta
PEAR	1.9.4	stable
PHPUnit	1.3.2	stable
PHP_Compat	1.5.0	stable
SOAP	0.13.0	beta
Structures_Graph	1.0.4	stable
Validate	0.8.5	beta
XML_RPC	1.5.5	stable
XML_Util	1.2.1	stable

接下来我们开始正式部署 Centreon 监控系统。

8.2.2 部署 Centreon 监控系统

首先需要从 Centreon 的官方网站上下载最新版本的 Centreon。本书选用的版本是 Centreon Stable version 版本 2.5.1，即图 8-5 中的 centreon-2.5.1。

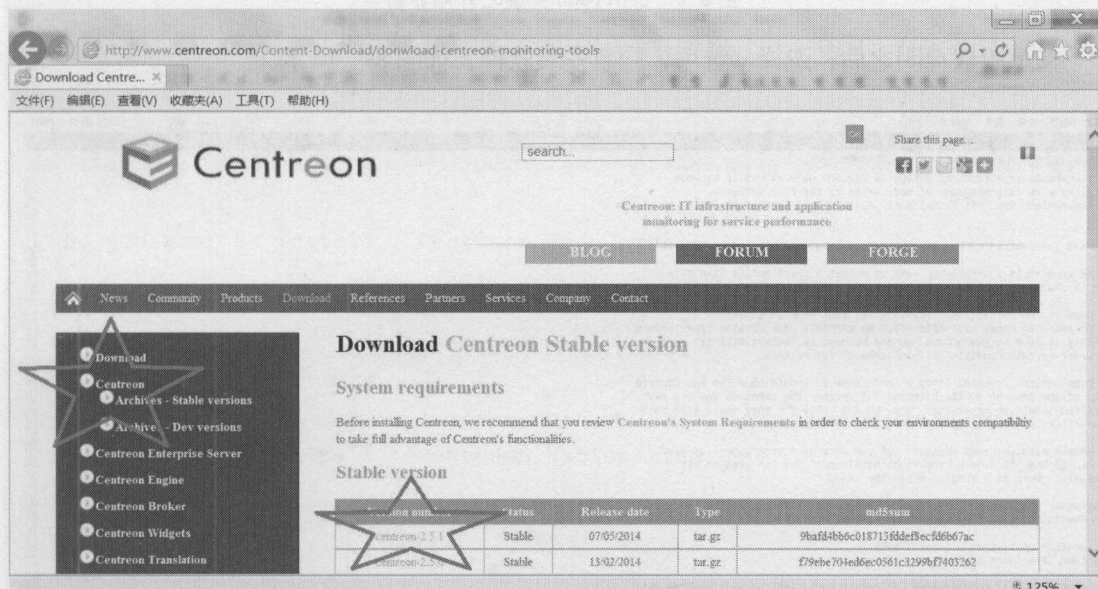


图 8-5 Centreon 下载网站

将下载的 centreon-2.5.1.tar.gz 文件上传至 Centreon 监控系统中央服务器上，执行下列语句进行解压缩。

```
#tar -zxvf centreon-2.5.1.tar.gz
#cd centreon-2.5.1
```

然后执行./install.sh -i 命令安装 Centreon，如图 8-6 所示。

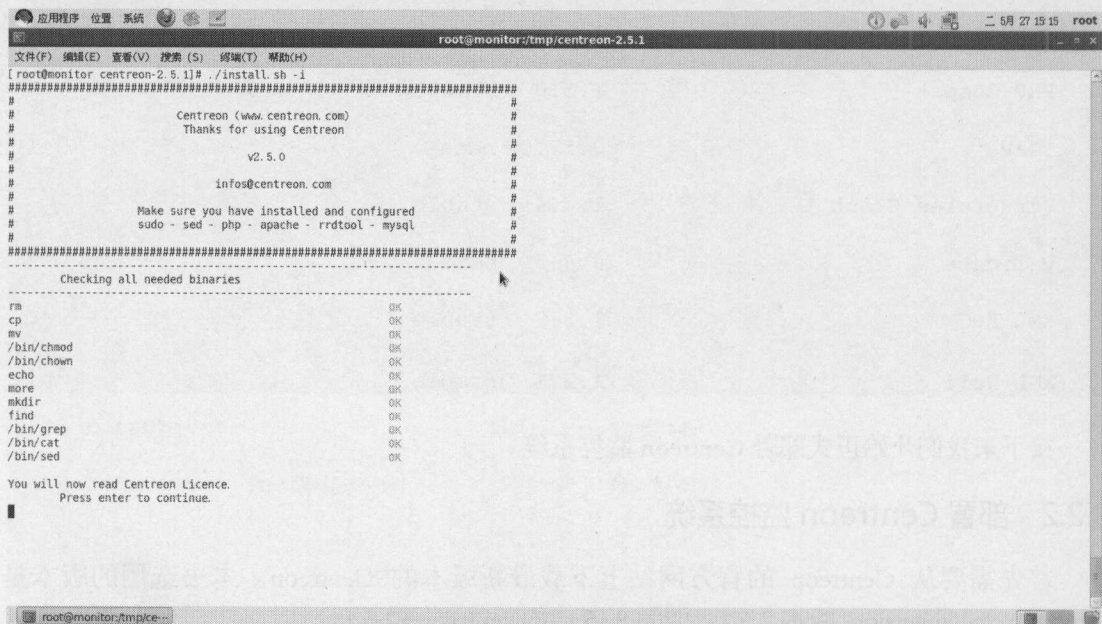


图 8-6 Centreon 安装之开始界面

阅读版权声明之后，选择 y 选项，同意安装软件，如图 8-7 所示。

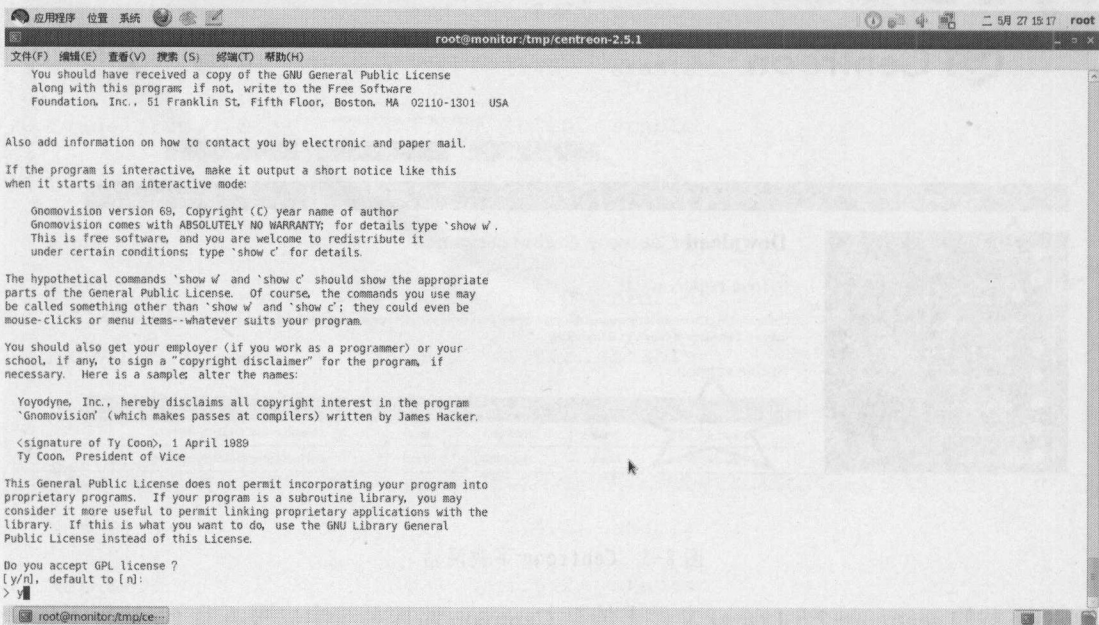


图 8-7 同意安装 Centreon

以下是文字版本的安装步骤，需要人工输入的步骤都用加粗字体显示，并适当配以文字说明。

```
This General Public License does not permit incorporating your program into
proprietary programs. If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library. If this is what you want to do, use the GNU Library General
Public License instead of this License.
```

```
Do you accept GPL license ?
```

```
[y/n], default to [n]:
```

```
> y
```

```
-----
Please choose what you want to install
-----
```

```
Do you want to install : Centreon Web Front
```

```
[y/n], default to [n]:
```

```
> y
```

```
Do you want to install : Centreon CentCore
```

```
[y/n], default to [n]:
```

```
> y
```

```
Do you want to install : Centreon Nagios Plugins
```

```
[y/n], default to [n]:
```

```
> y
```

```
Do you want to install : Centreon Snmp Traps process
```

```
[y/n], default to [n]:
```

```
> y
```



```
-----  
Start CentWeb Installation  
-----
```

Where is your Centreon directory?

default to [/usr/local/centreon]

>

Path **/usr/local/centreon**

OK

Where is your Centreon log directory

default to [/usr/local/centreon/log]

>

Path **/usr/local/centreon/log**

OK

Where is your Centreon etc directory

default to [/etc/centreon]

>

Path **/etc/centreon**

OK

Where is your Centreon binaries directory

default to [/usr/local/centreon/bin]

>

Path **/usr/local/centreon/bin**

OK

Where is your Centreon data informations directory

default to [/usr/local/centreon/data]

>

Path **/usr/local/centreon/data**

OK

Where is your Centreon variable library directory?

default to [/var/lib/centreon]

>

Path **/var/lib/centreon**

OK

/usr/bin/rrdtool

OK

/bin/mail

OK

/usr/bin/php

OK

Where is PEAR [PEAR.php]

default to [/usr/share/php/PEAR.php]

>

/usr/share/php/PEAR.php is not a valid file.

CRITICAL

Where is PEAR [PEAR.php]

default to [/usr/share/php/PEAR.php]

> **/usr/share/pear/PEAR.php**

Path /usr/share/pear

OK

/usr/bin/perl

OK

Finding Apache user :

apache

Finding Apache group :

apache

What is the Centreon group ? [centreon]

default to [centreon]

>

Do you want me to create this group ? [centreon]

[y/n], default to [n]:

> **y**

What is the Centreon user ? [centreon]


```
default to [centreon]

>

Do you want me to create this user ? [centreon]

[y/n], default to [n]:

> y

What is the Monitoring engine user ?

>

The user not_def does not exists.                                CRITICAL

What is the Monitoring engine user ?

> nagios

What is the Broker user ? (optional)

> nagios

What is the Monitoring engine log directory ?

> /usr/local/nagios/var

Where is your monitoring plugins (libexec) directory ?

default to [/usr/lib/nagios/plugins]

> /usr/local/nagios/libexec

Path /usr/local/nagios/libexec                                    OK

Add group centreon to user apache                                OK

Add group centreon to user nagios                                OK

Add group nagios to user apache                                  OK

Add group nagios to user centreon                                OK

-----

Configure Sudo
```

Where is sudo configuration file

default to [/etc/sudoers]

>

/etc/sudoers

OK

What is the Monitoring engine init.d script ?

> **/etc/init.d/nagios**

What is the Monitoring engine binary ?

> **/usr/local/nagios/bin/nagios**

What is the Monitoring engine configuration directory ?

> **/usr/local/nagios/etc**

Where is the configuration directory for broker module ?

> **/usr/local/nagios/etc**

Where is the init script for broker module daemon ?

> **/etc/init.d/ndo2db**

/etc/init.d/ndo2db is not a valid file.

CRITICAL

Where is the init script for broker module daemon ?

> **/etc/init.d/ndo2db**

Your sudo is not configured

Do you want me to configure your sudo ? (WARNING)

[y/n], default to [n]:

>

Please configure your sudo with this example:

/usr/local/centreon/examples/centreon.sudo

PASSED

Configure Apache server

Do you want to add Centreon Apache sub configuration file ?

[y/n], default to [n]:

> y

Create '/etc/httpd/conf.d/centreon.conf' OK

Configuring Apache OK

Do you want to reload your Apache ?

[y/n], default to [n]:

> y

Reloading Apache service FAIL

Preparing Centreon temporary files

Change right on /usr/local/centreon/log OK

Change right on /etc/centreon OK

Change macros for insertBaseConf.sql OK

Change macros for sql update files OK

Change macros for php files OK

Change macros for perl binary OK

Change right on /usr/local/nagios/etc OK

Add group nagios to user apache OK

Add group nagios to user nagios OK

Add group centreon to user nagios OK

Copy CentWeb in system directory

Install CentWeb (web front of centreon) OK

Change right for install directory

Change right for install directory OK

Install libraries OK

Write right to Smarty Cache	OK
Copying libinstall	OK
Change macros for centreon.cron	OK
Install Centreon cron.d file	OK
Change macros for centAcl.php	OK
Change macros for downtimeManager.php	OK
Install cron directory	OK
Change right for eventReportBuilder	OK
Change right for dashboardBuilder	OK
Change macros for centreon.logrotate	OK
Install Centreon logrotate.d file	OK
Prepare centFillTrapDB	OK
Install centFillTrapDB	OK
Prepare centreon_trap_send	OK
Install centreon_trap_send	OK
Prepare centreon_check_perfdata	OK
Install centreon_check_perfdata	OK
Prepare centreonSyncPlugins	OK
Install centreonSyncPlugins	OK
Prepare centreonSyncArchives	OK
Install centreonSyncArchives	OK
Install generateSqlLite	OK
Install changeRrdDsName.pl	OK
Prepare export-mysql-indexes	OK
Install export-mysql-indexes	OK
Prepare import-mysql-indexes	OK
Install import-mysql-indexes	OK
Centreon Web Perl lib installed	OK


```
-----
Pear Modules
-----
```

```
Check PEAR modules
```

PEAR	1.4.9	1.9.4	OK
DB	1.7.6	1.7.14	OK
DB_DataObject	1.8.4	1.11.3	OK
DB_DataObject_FormBuilder	1.0.0RC4	1.0.2	OK
MDB2	2.0.0	2.4.1	OK
Date	1.4.6	1.4.7	OK
HTML_Common	1.2.2	1.2.5	OK
HTML_QuickForm	3.2.5	3.2.13	OK
HTML_QuickForm_advmultiselect	1.1.0	1.5.1	OK
HTML_Table	1.6.1	1.8.3	OK
Archive_Tar	1.1	1.3.11	OK
Auth_SASL	1.0.1	1.0.6	OK
Console_Getopt	1.2	1.3.1	OK
Net_SMTP	1.2.8	1.6.2	OK
Net_Socket	1.0.1	1.0.14	OK
Net_Traceroute	0.21	0.21.3	OK
Net_Ping	2.4.1	2.4.5	OK
Validate	0.6.2	0.8.5	OK
XML_RPC	1.4.5	1.5.5	OK
SOAP	0.10.1	0.13.0	OK
Log	1.9.11	1.12.7	OK
Archive_Zip	0.1.2		NOK

```
Do you want me to install/upgrade your PEAR modules
```

```
[y/n], default to [y]:
```

> y

Upgrading PEAR modules

Installing PEAR modules

Archive_Zip	0.1.2	0.1.2	OK
-------------	-------	-------	----

Check PEAR modules

PEAR	1.4.9	1.9.4	OK
------	-------	-------	----

DB	1.7.6	1.7.14	OK
----	-------	--------	----

DB_DataObject	1.8.4	1.11.3	OK
---------------	-------	--------	----

DB_DataObject_FormBuilder	1.0.0RC4	1.0.2	OK
---------------------------	----------	-------	----

MDB2	2.0.0	2.4.1	OK
------	-------	-------	----

Date	1.4.6	1.4.7	OK
------	-------	-------	----

HTML_Common	1.2.2	1.2.5	OK
-------------	-------	-------	----

HTML_QuickForm	3.2.5	3.2.13	OK
----------------	-------	--------	----

HTML_QuickForm_advmultiselect	1.1.0	1.5.1	OK
-------------------------------	-------	-------	----

HTML_Table	1.6.1	1.8.3	OK
------------	-------	-------	----

Archive_Tar	1.1	1.3.11	OK
-------------	-----	--------	----

Auth_SASL	1.0.1	1.0.6	OK
-----------	-------	-------	----

Console_Getopt	1.2	1.3.1	OK
----------------	-----	-------	----

Net_SMTP	1.2.8	1.6.2	OK
----------	-------	-------	----

Net_Socket	1.0.1	1.0.14	OK
------------	-------	--------	----

Net_Traceroute	0.21	0.21.3	OK
----------------	------	--------	----

Net_Ping	2.4.1	2.4.5	OK
----------	-------	-------	----

Validate	0.6.2	0.8.5	OK
----------	-------	-------	----

XML_RPC	1.4.5	1.5.5	OK
---------	-------	-------	----

SOAP	0.10.1	0.13.0	OK
------	--------	--------	----

Log	1.9.11	1.12.7	OK
-----	--------	--------	----

Archive_Zip	0.1.2	0.1.2	OK
-------------	-------	-------	----

All PEAR modules			OK
------------------	--	--	----

Centreon Post Install

Create /usr/local/centreon/www/install/install.conf.php OK

Create /etc/centreon/instCentWeb.conf OK

Start CentStorage Installation

Where is your Centreon Run Dir directory?

default to [/var/run/centreon]

>

Do you want me to create this directory ? [/var/run/centreon]

[y/n], default to [n]:

> y

Path /var/run/centreon OK

Where is your CentStorage binary directory

default to [/usr/local/centreon/bin]

>

Path /usr/local/centreon/bin OK

Where is your CentStorage RRD directory

default to [/var/lib/centreon]

>

Path /var/lib/centreon OK

Preparing Centreon temporary files

/tmp/centreon-setup exists, it will be moved...


```

install www/install/createTablesCentstorage.sql          OK
Creating Centreon Directory '/var/lib/centreon/status'    OK
Creating Centreon Directory '/var/lib/centreon/metrics'   OK
Install CentStorage binary                                OK
Change right : /var/run/centreon                           OK
Change macros for centstorage init script                 OK
Replace CentStorage sysconfig script Macro                OK

Do you want me to install CentStorage init script ?
[y/n], default to [n]:

> y

CentStorage init script installed                          OK
CentStorage sysconfig script installed                     OK

Do you want me to install CentStorage run level ?
[y/n], default to [n]:

> y

CentStorage Perl lib installed                             OK
Install logAnalyser                                        OK
Install logAnalyserBroker                                 OK
Install nagiosPerfTrace                                    OK
Change macros for centstorage.cron                         OK
Install CentStorage cron                                   OK
Change macros for centstorage.logrotate                   OK
Install Centreon Storage logrotate.d file                 OK
Create /etc/centreon/instCentStorage.conf                 OK

```

Start CentCore Installation

```
-----

Where is your CentCore binary directory
default to [/usr/local/centreon/bin]

>

Path /usr/local/centreon/bin                                OK

Preparing Centreon temporary files

/tmp/centreon-setup exists, it will be moved...

Copy CentCore in binary directory                            OK

Change right : /var/run/centreon                             OK

Change right : /var/lib/centreon                             OK

Change macros for centcore.logrotate                         OK

Install Centreon Core logrotate.d file                       OK

Replace CentCore init script Macro                           OK

Replace CentCore sysconfig script Macro                      OK


Do you want me to install CentCore init script ?

[y/n], default to [n]:

> y

CentCore init script installed                                OK

CentCore sysconfig script installed                           OK


Do you want me to install CentCore run level ?

[y/n], default to [n]:

> y

CentCore Perl lib installed                                   OK

Create /etc/centreon/instCentCore.conf                       OK

-----

Start CentPlugins Installation
```

Where is your CentPlugins lib directory

default to [/var/lib/centreon/centplugins]

>

Do you want me to create this directory ? [/var/lib/centreon/centplugins]

[y/n], default to [n]:

> **y**

Path /var/lib/centreon/centplugins

OK

Preparing Centreon temporary files

/tmp/centreon-setup exists, it will be moved...

Change macros for CentPlugins

OK

Installing the plugins

OK

Change right on centreon.conf

OK

CentPlugins is installed

Start CentPlugins Traps Installation

Where is your SNMP configuration directory

default to [/etc/snmp]

>

/etc/snmp

OK

Where is your CentreonTrapd binaries directory

default to [/usr/local/centreon/bin]

>

/usr/local/centreon/bin

OK


```
Finding Apache user :                               apache

Preparing Centreon temporary files

/tmp/centreon-setup exists, it will be moved...

Change macros for snmptrapd.conf                     OK

Replace CentreonTrapd init script Macro              OK

Replace CentreonTrapd sysconfig script Macro         OK


Do you want me to install CentreonTrapd init script ?

[y/n], default to [n]:

> y

CentreonTrapd init script installed                  OK

CentreonTrapd sysconfig script installed             OK


Do you want me to install CentreonTrapd run level ?

[y/n], default to [n]:

> y

trapd Perl lib installed                            OK

Install : snmptrapd.conf                             OK

Install : centreontrapdforward                       OK

Install : centreontrapd                             OK

Create /etc/centreon/instCentPlugins.conf           OK

#####

#                                                    #

#           Go to the URL : http://localhost/centreon/           #

#           to finish the setup                     #

#                                                    #

#           Report bugs at http://forge.centreon.com           #

#                                                    #

#           Thanks for using Centreon.               #
```

```

# -----#
# Contact : infos@centreon.com#
# http://www.centreon.com#
#
#####

```

至此，Centreon 监控软件在操作系统上安装成功，下面进入安装后的配置步骤。

8.3 安装后配置

打开 IE 或者 Mozilla Firefox 浏览器，输入网址 <http://ip/centreon>，打开 Centreon 配置界面，如图 8-8 所示。

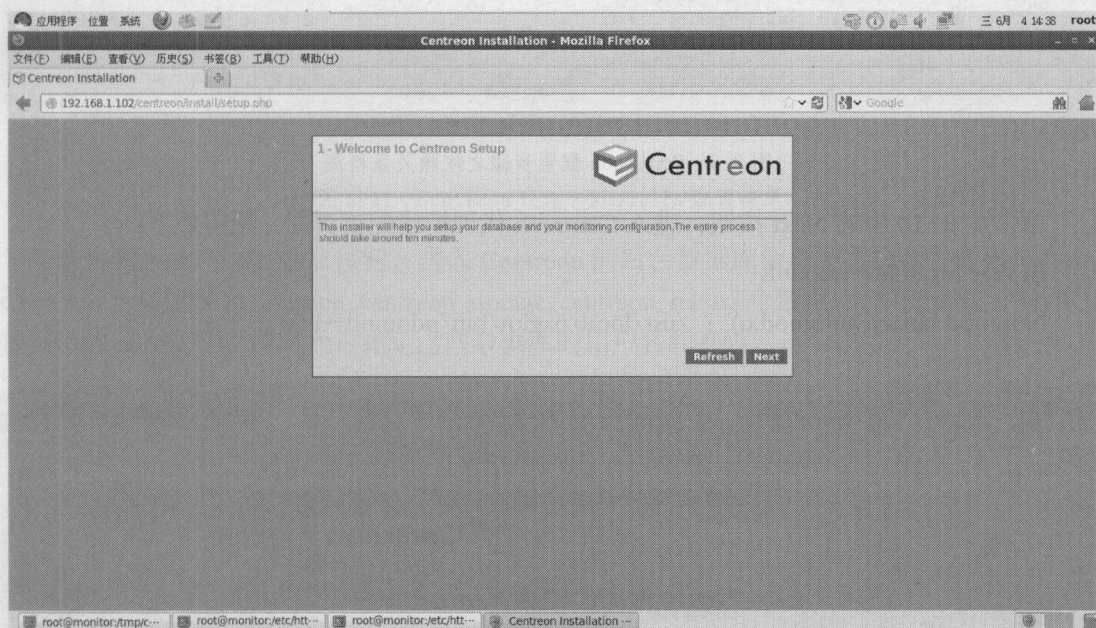


图 8-8 进入 Centreon 配置界面

单击图 8-8 的 Next 按钮，进入依赖软件检测界面，如图 8-9 所示，依赖软件检测应该全部是 Loaded 状态，否则就需要进一步检查并安装缺失的依赖软件。

单击图 8-9 中的 Next 按钮，进入监控引擎——Nagios 的配置页面，各项明细如下：

Nagios directory : /usr/local/nagios

Nagiostats binary : /usr/local/nagios/bin/nagiostats

Nagios image directory : /usr/local/nagios/share/images

Embedded Perl initialisation file : /usr/local/nagios/bin/p1.pl

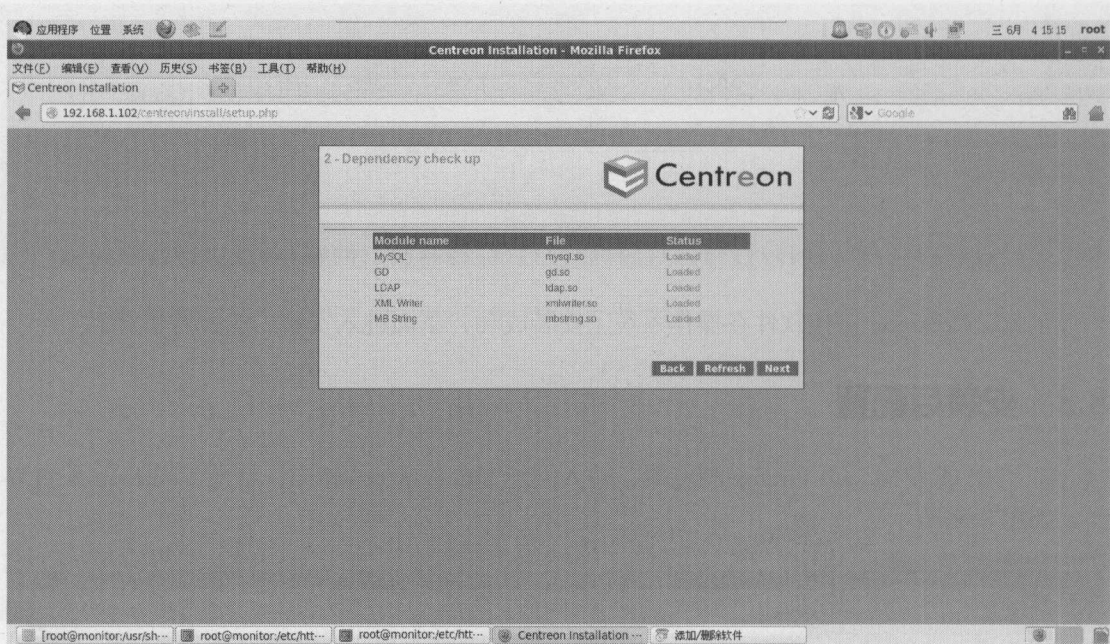


图 8-9 Centreon 配置步骤之依赖关系检测

单击图 8-10 中的 Next 按钮，进入 Centreon 代理模块配置页面，配置如下：

Broker Module：ndoutils

Ndomod binary (ndomod.o)：/usr/local/nagios/bin/ndomod-3x.o

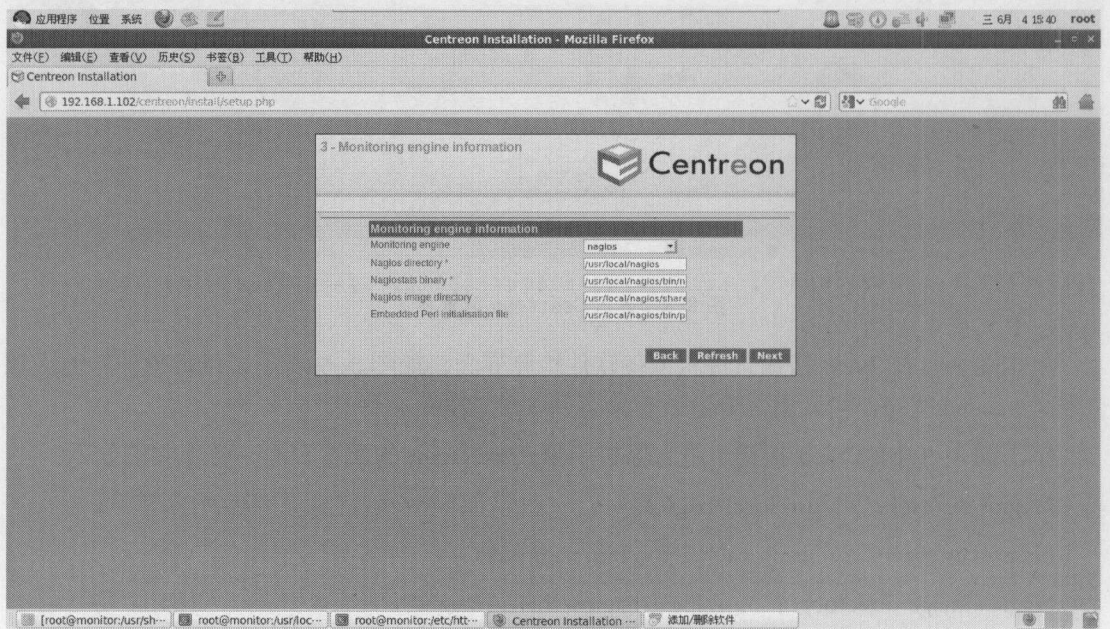


图 8-10 Centreon 监控引擎—Nagios 的配置页面

单击图 8-11 中的 Next 按钮，进入管理员信息配置页面，输入合适的密码和联系邮件。

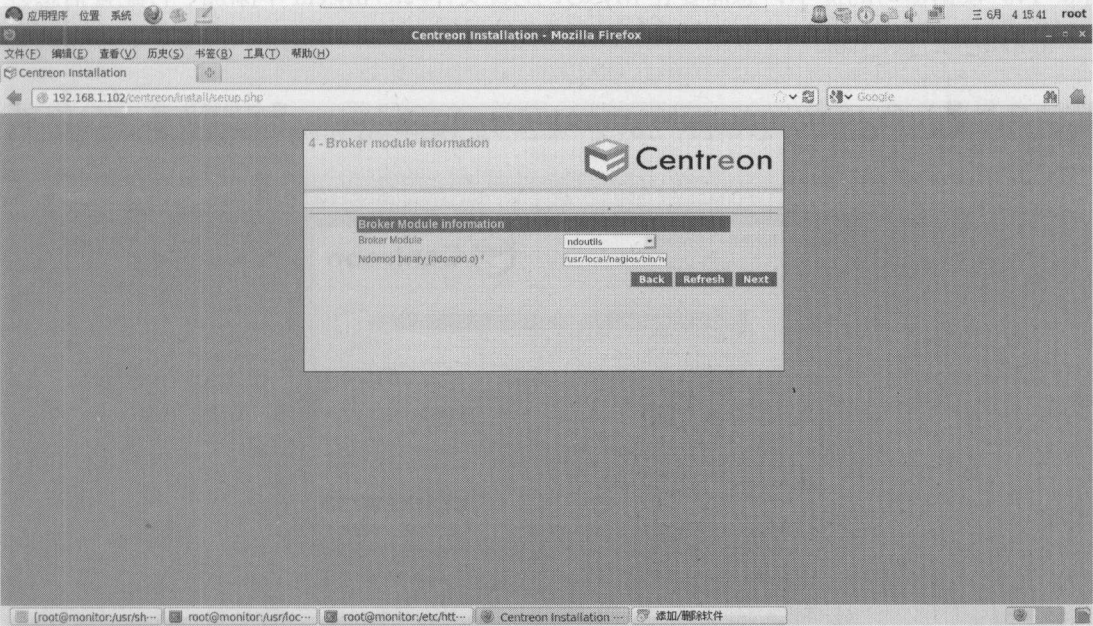


图 8-11 Centreon 代理—NDOutils 配置信息

单击图 8-12 中的 Next 按钮，进入 Centreon 的后台数据库信息配置页面。在该页面中，Centreon 会创建名为 centreon、centreon_storage、centreon_status3 个数据库，前提是知道 MySQL 数据库的 root 密码，同时允许指定这些数据库的用户名和密码。

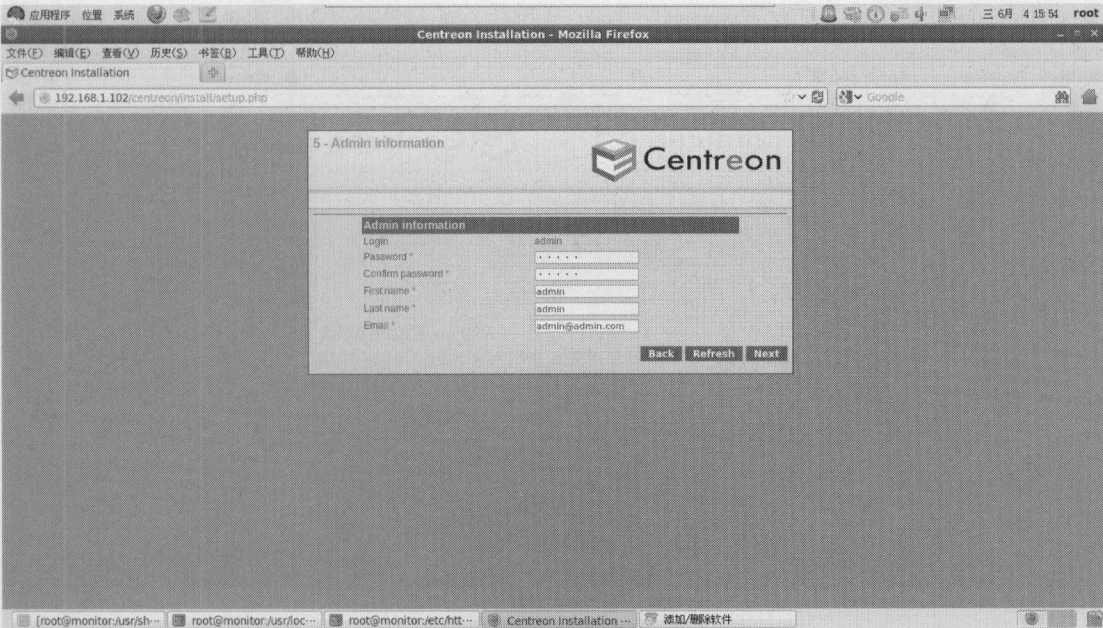


图 8-12 Centreon 管理员信息配置

单击图 8-13 中的 Next 按钮，开始 Centreon 的后台数据库创建及对象初始化步骤，如图 8-14。根据窗口的提示，您需要在 MySQL 配置文件/etc/my.cnf 中添加一行配置信息：

```
innodb_file_per_table=1
```

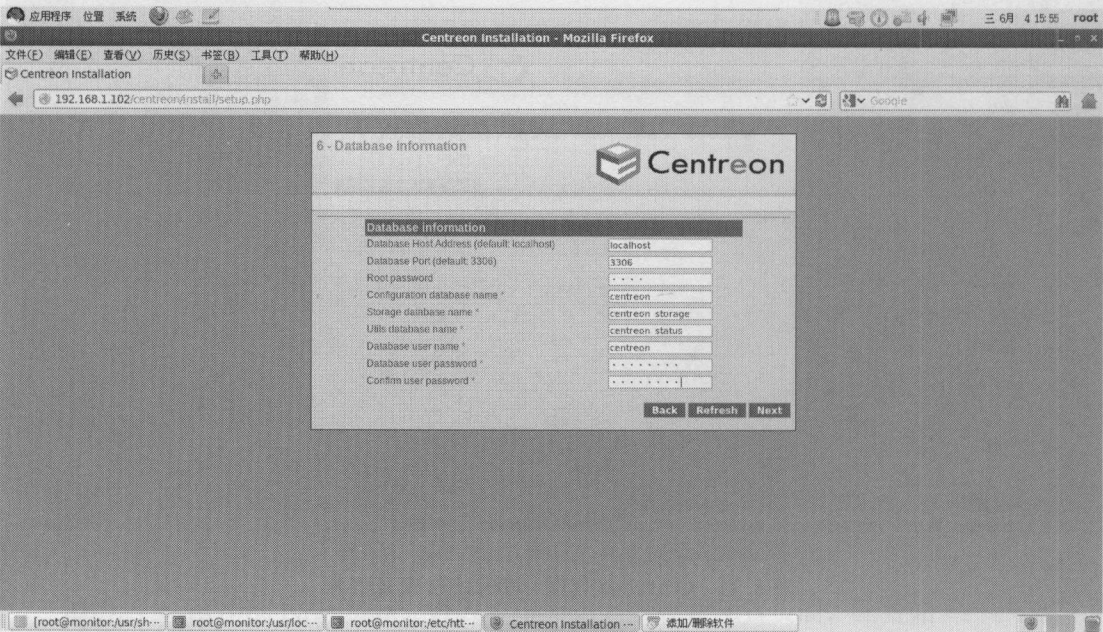


图 8-13 Centreon 后台数据库配置

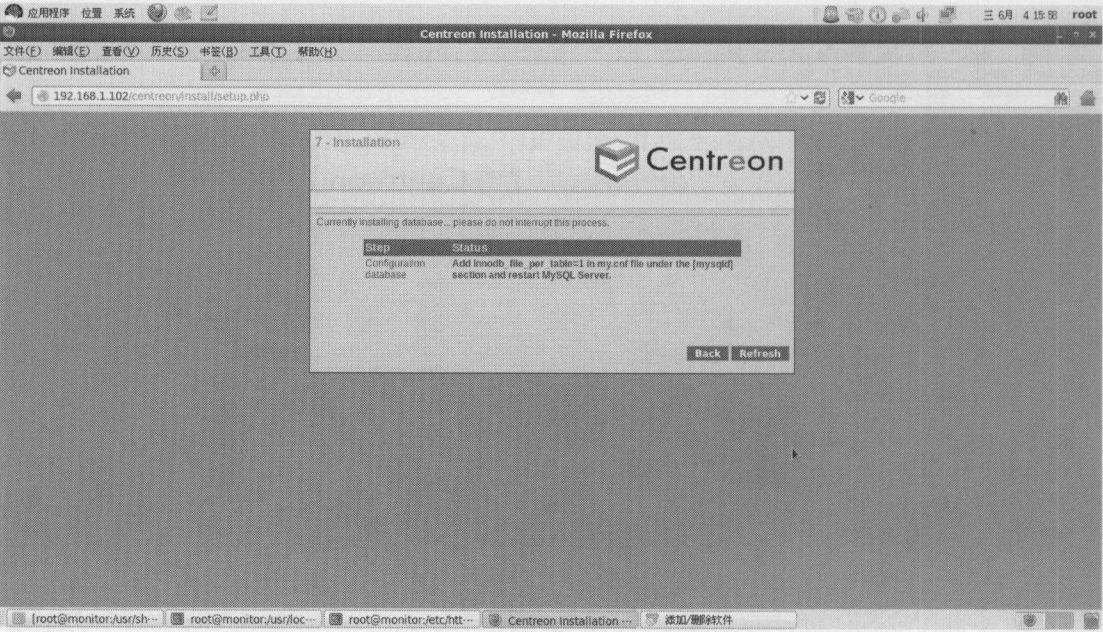


图 8-14 Centreon 后台数据库初始化

添加 MySQL 配置信息，并重启 MySQL 数据库，如图 8-15 所示。

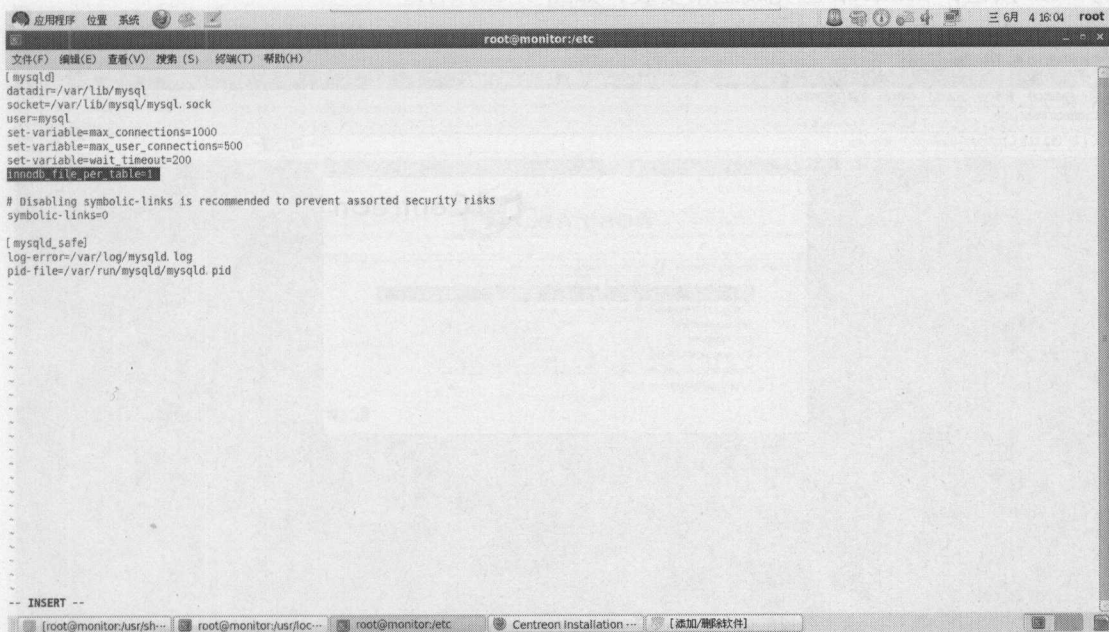


图 8-15 Centreon 数据库安装——修改 MySQL 配置文件

MySQL 数据库重启完毕后，单击图 8-14 中的 Refresh 按钮，开始进入 Centreon 后台数据库初始化界面，如图 8-16 所示。

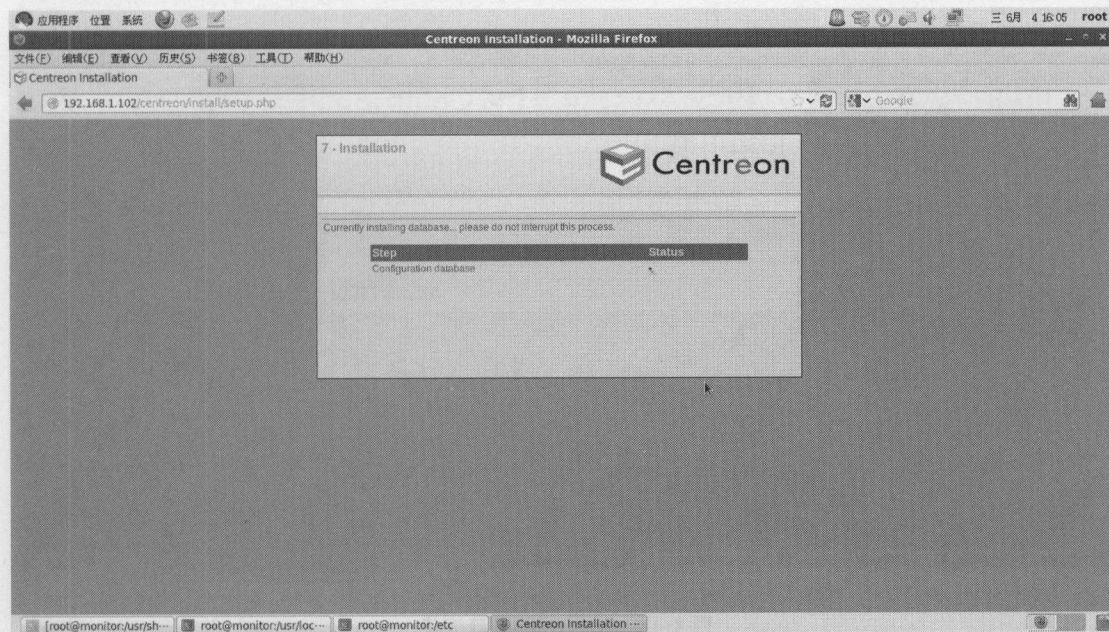


图 8-16 Centreon 数据库后台正在初始化

添加 MySQL 配置信息，并重启 MySQL 数据库，如图 8-15 所示。

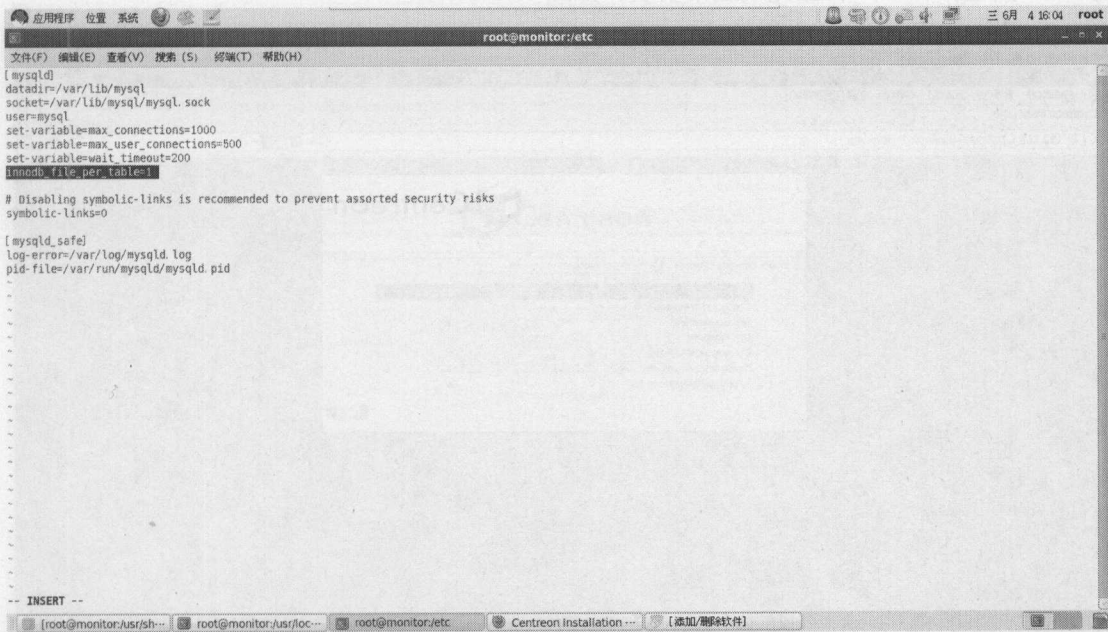


图 8-15 Centreon 数据库安装——修改 MySQL 配置文件

MySQL 数据库重启完毕后，单击图 8-14 中的 Refresh 按钮，开始进入 Centreon 后台数据库初始化界面，如图 8-16 所示。

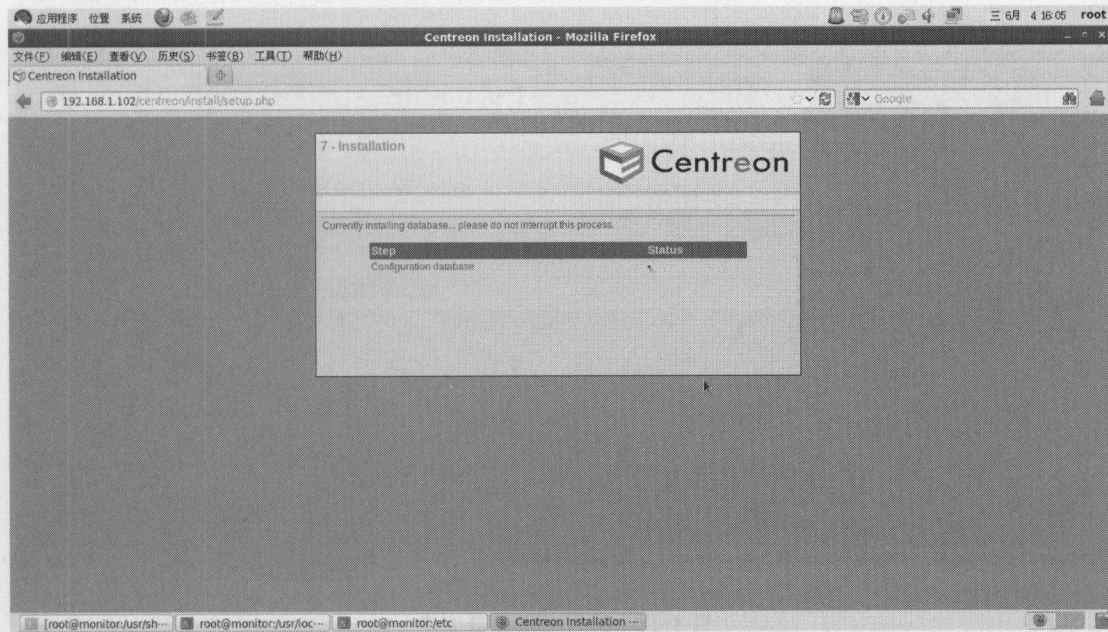


图 8-16 Centreon 数据库后台正在初始化

单击图 8-18 中的 Finish 按钮，进入 Centreon 的登录界面，如图 8-19 所示。



图 8-19 Centreon 首次登录界面

8.4 Centreon 的 Web 用户界面

Centreon 的默认主页如图 8-20 所示。

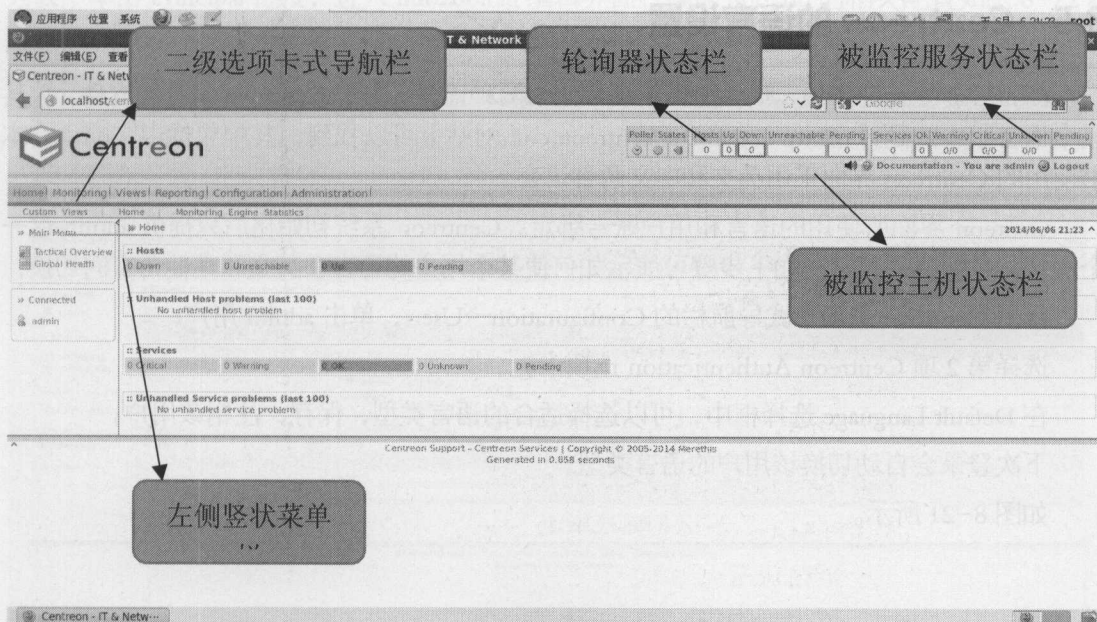


图 8-20 Centreon 的默认主页

Centreon 监控系统的默认主页由固定的功能分区构成。在页面上方的“Poller States（轮

询器状态栏)”功能区中，依次显示了系统的检测延迟、检测命令是否正常，以及检测命令与数据库之间的读写是否正常共 3 个图标。在页面右上方的“被监控主机状态栏”和“被监控服务状态栏”监控区域，可以根据告警信息的严重程度显示被监控服务器或者监控项的实时告警信息，同时点击这些数字，可快速跳转至根据相应条件过滤后的监控项列表中。在功能区的下方，显示了目前登陆系统的用户账号，以及注销按钮。

接下来的是典型的 Centreon 两级选项卡式导航栏界面。第一级导航栏共有 6 个选项卡，前 4 个主要是查看、监控功能，适合操作人员使用，后 2 个主要是配置功能，适合监控平台系统管理员使用。

- **Home（默认主页）**选项卡主要显示各类自定义或者默认的监控视图，以及调度进程（Nagios 或者 Centreon Broker 进程）的各类统计信息。
- **Monitoring（实时监控）**选项卡显示各类实时告警信息，并提供对于告警信息的查找、过滤、处理等功能。
- **Views（视图查看）**选项卡提供告警图形及日志查看功能。
- **Reporting（报告与报表）**选项卡提供了在指定监控区间段内，简单的监控报表功能。
- **Configuration（系统配置）**选项卡提供 Centreon 监控平台的配置管理功能，通常该选项卡只有最高权限用户才能进入。
- **Administration（系统管理）**主要供系统管理员进行监控平台的核心配置调整、参数调整以及权限管理等工作，只有最高权限用户才能进入。

以上选项卡都可以通过调整权限的方式选择为隐藏或者显示，便于不同权限的用户访问。详细的选项卡功能信息在后续的章节中会有介绍。

8.5 Centreon 的语言设置

Centreon 安装完毕后，默认提供英语和法语两种语言。由于 Centreon 是开源软件，故其多个语言版本的翻译在<http://translate.centreon.com>网站上可以找到，其中包括中文翻译。本书使用的 Centreon 界面采用英文和中文两种语言。

Centreon 界面所使用的语言和用户账号绑定。Centreon 系统初始化后只提供 admin 这一个账号，且默认是英文。如下步骤可演示如何使某个用户的语言在英文和其他语言间切换：

单击 Centreon 选项卡式导航栏的 Configuration→Users，单击 admin 用户；

选择第 2 项 Centreon Authentication 选项卡；

在 Default Language 选择框中，可以选择适合的语言类型，保存，注销该用户；

下次登录会自动切换该用户的语言类型。

如图 8-21 所示。

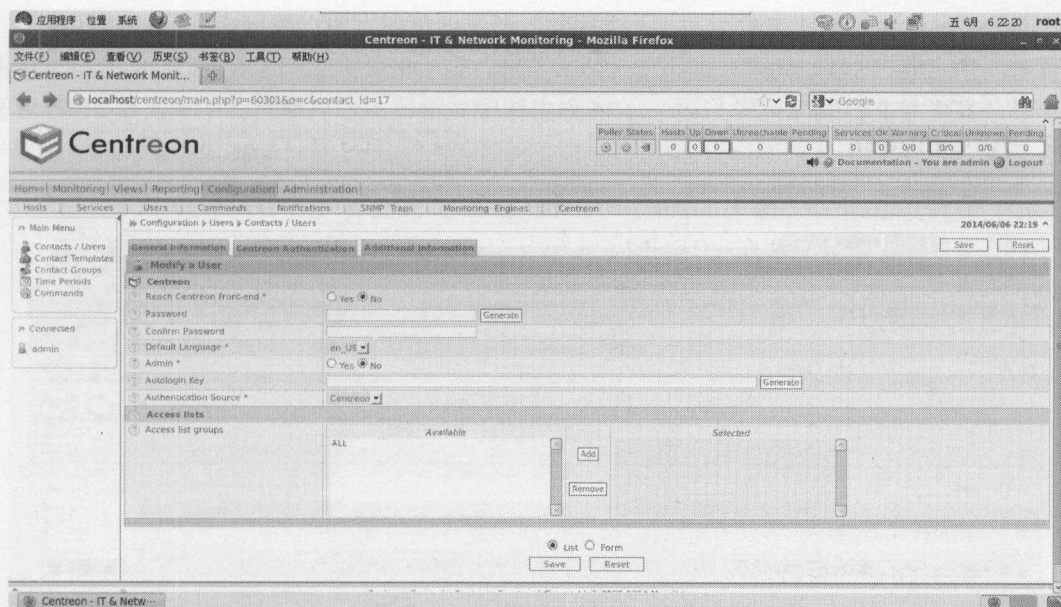


图 8-21 Centreon Web 界面用户语言设置

8.6 Centreon 的数据库连接配置

安装完毕后, 我们还要在 Centreon 界面中对 NDODB 和 NDOUtils 做相关配置, 方能正确访问并展示 Nagios 采集的监控数据。

单击 Configuration→Centreon 导航菜单, 再单击左侧菜单栏的 ndo2db.cfg 链接, 接着在列表中单击 Principal 链接, 进入 ndo2db.cfg 编辑界面, 确保 ndo2db 的相关配置如图 8-22 和图 8-23 所示, 单击 Save 按钮保存相关配置。

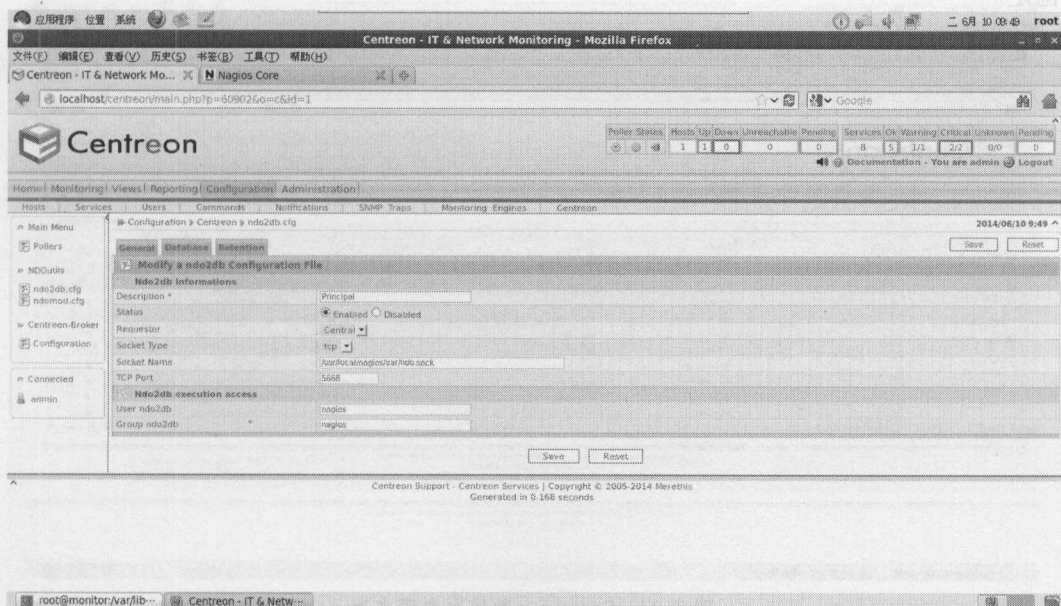


图 8-22 在 Centreon 中编辑 ndo2db 配置-General1

为了 Centreon 中配置的被监控服务器能够部署在 Nagios (参考图 7-4 centcore 功能架构图中 Centreon 与 Nagios 的协作关系) 中, 接下来我们需要在 Centreon 用户界面中, 将存放在 MySQL 数据库中的被监控服务器相关配置生成 Nagios 可以识别的配置文件。

遵循以下简单步骤:

- (1) 进入 Centreon 用户界面的 Configuration→Monitoring Engine 菜单, 选择 Generate Configuration Files 和 Run monitoring engine debug (-v) 两项配置, 如图 8-25 所示。

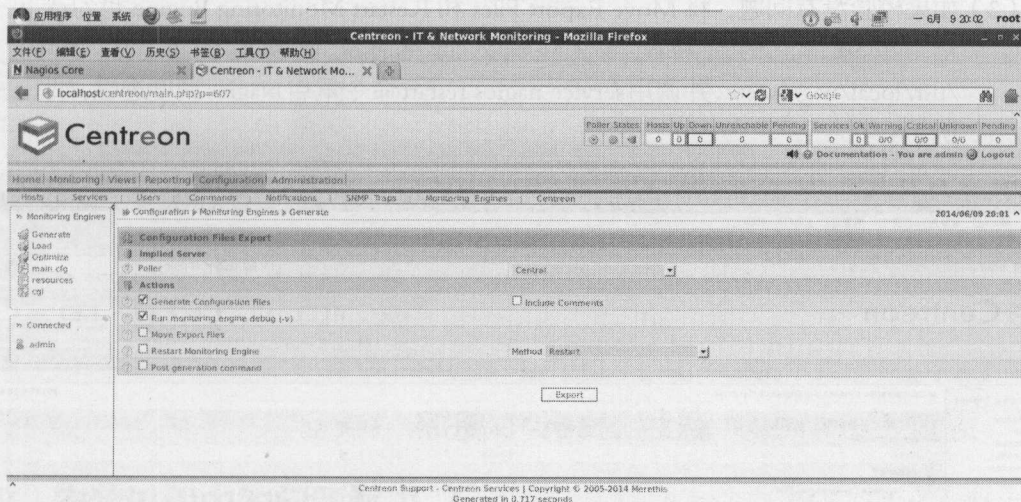


图 8-25 Centreon 导出验证

- (2) 单击图 8-25 中的 Export 按钮, 进入如图 8-26 界面。

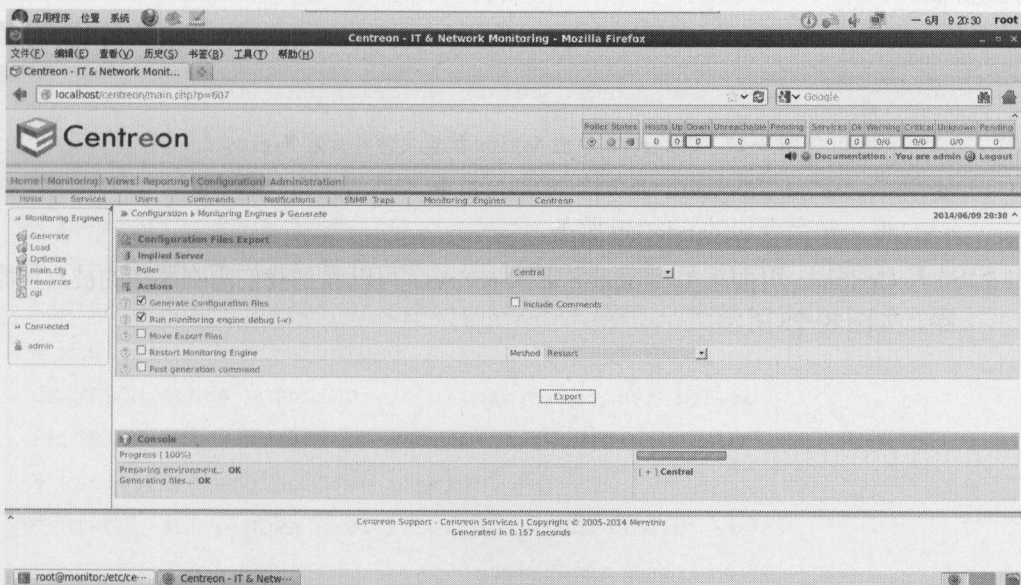


图 8-26 Centreon 成功导出 Nagios 配置文件

此处 Export 的作用就是使用调用 Nagios 的 `nagios -v` 命令进行配置校验,检查由 Centreon 配置并生成的 Nagios 配置文件是否正确,是否能够被 Nagios 识别。如果此校验步骤出问题,那么就需要进一步检查 Centreon 中被监控服务器或者被监控项的配置,尤其是重点检查最近刚刚做过的监控项变更。在 Centreon 的配置过程中,存在多项琐碎的操作,且每一步操作都会提交到后台 MySQL 数据库中,无法自动回退,因此记好每一步操作至关重要,以便于及时手工回退。

- (3) 如果校验没有问题,将 Move Export Files 和 Restart Monitoring Engine 也勾选上,执行 Export,此时会将配置文件写入到 Nagios 的实际的配置目录中,即 `/usr/local/nagios/etc`,并调用 `service nagios restart` 命令重启 Nagios 调度进程,如图 8-27 所示。

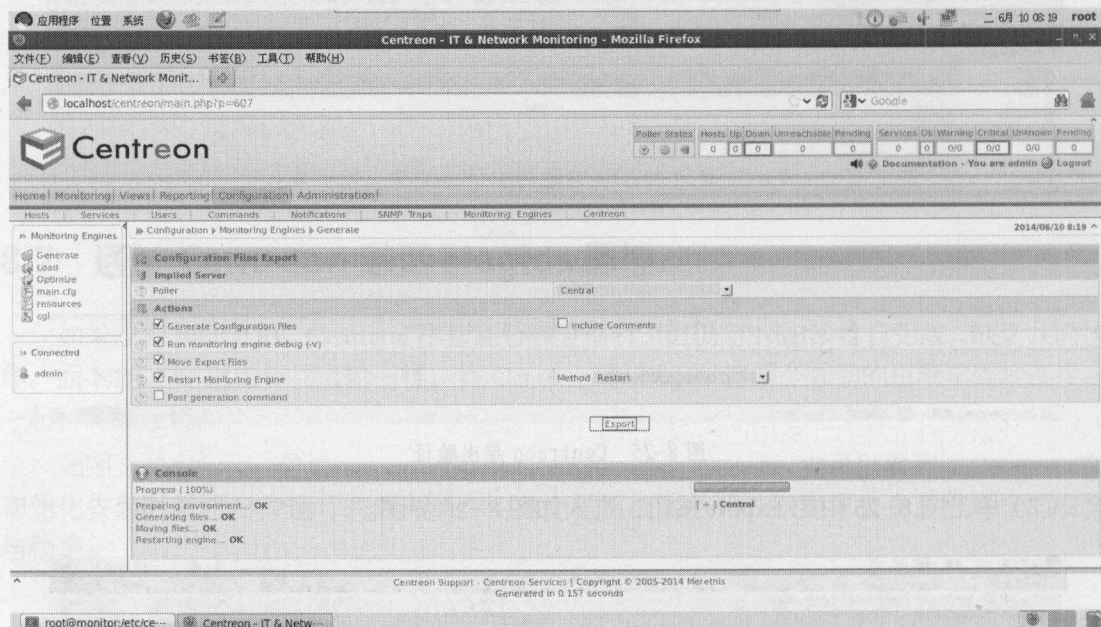


图 8-27 Centreon 导出 Nagios 配置文件并重启 Nagios

- (4) 使用 `service ndo2db status` 命令检查 ndo2db 服务是否正常运行,如果没有,使用 `service ndo2db start` 命令启动 ndo2db 服务。
- (5) 进入 Centreon 页面菜单 Monitoring→Services, 可以看到我们刚刚添加的被监控服务器,如图 8-28 所示。

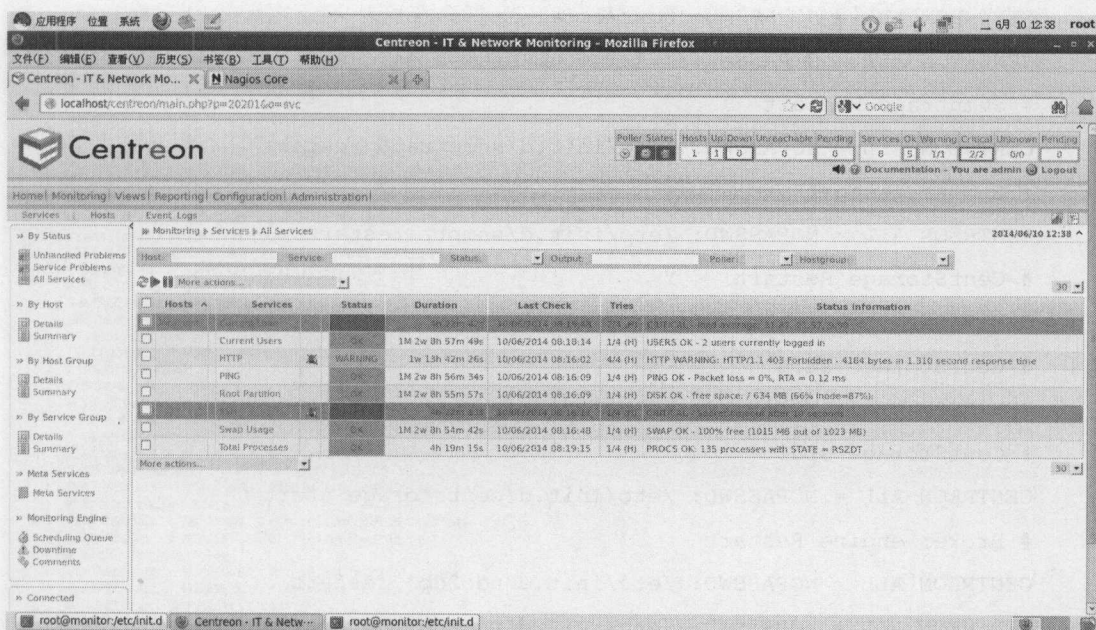


图 8-28 Centreon 中查看被监控服务器

8.8 安装过程中的问题解决

8.8.1 Export 时显示 sudo 相关错误

在图 8-26 中导出 Nagios 配置文件时,如果单击 Export 按钮出现 sudo: no tty present and no askpass program specified 相关错误,需要在 root 用户下执行 visudo 命令,编辑 sudo 配置,添加如下的配置信息。

```
#Add by CENTREON installation script
User_Alias CENTREON=apache,centreon
Defaults:CENTREON !requiretty
# Monitoring engine Restart
CENTREON ALL = NOPASSWD: /etc/init.d/nagios* restart
CENTREON ALL = NOPASSWD: /etc/init.d/nagios restart
# Monitoring engine reload
CENTREON ALL = NOPASSWD: /etc/init.d/nagios* reload
CENTREON ALL = NOPASSWD: /etc/init.d/nagios reload
# Monitoring engine test config
CENTREON ALL = NOPASSWD: /usr/local/nagios/etc* -v *
CENTREON ALL = NOPASSWD: /usr/local/nagios/etc -v *
# Monitoring engine test for optim config
```



```

CENTREON ALL = NOPASSWD: /usr/local/nagios/etc* -s *
CENTREON ALL = NOPASSWD: /usr/local/nagios/etc -s *
# Snmptrapd Restart
CENTREON ALL = NOPASSWD: /etc/init.d/snmptrapd restart
# Snmptt restart
CENTREON ALL = NOPASSWD: /etc/init.d/snmptt restart
# CentStorage Restart
CENTREON ALL = NOPASSWD: /etc/init.d/centstorage restart
# CentStorage stop
CENTREON ALL = NOPASSWD: /etc/init.d/centstorage stop
# CentStorage start
CENTREON ALL = NOPASSWD: /etc/init.d/centstorage start
# Broker engine Restart
CENTREON ALL = NOPASSWD: /etc/init.d/ndo2db* restart
CENTREON ALL = NOPASSWD: /etc/init.d/ndo2db restart
# Broker engine reload
CENTREON ALL = NOPASSWD: /etc/init.d/ndo2db* reload
CENTREON ALL = NOPASSWD: /etc/init.d/ndo2db reload
# Monitoring engine test config
CENTREON ALL = NOPASSWD: /usr/local/nagios/bin/nagios* -v *
CENTREON ALL = NOPASSWD: /usr/local/nagios/bin/nagios -v *
# Monitoring engine test for optim config
CENTREON ALL = NOPASSWD: /usr/local/nagios/bin/nagios* -s *
CENTREON ALL = NOPASSWD: /usr/local/nagios/bin/nagios -s *
## END: CENTREON SUDO

```

如图 8-29 所示:

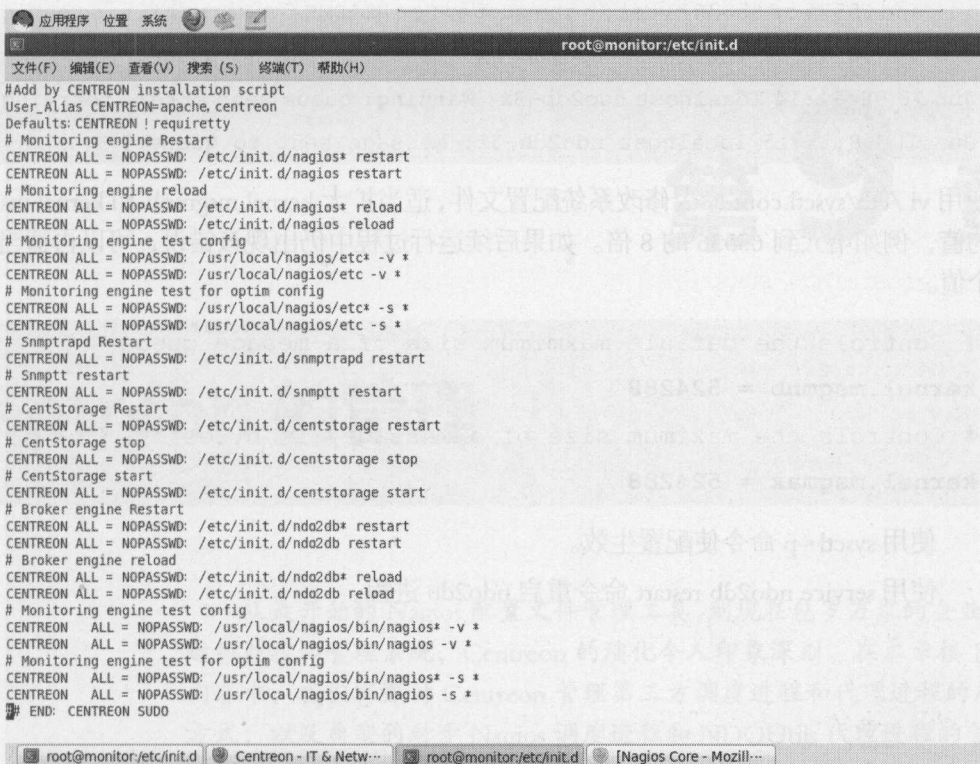


图 8-29 visudo 配置

同时进入 Centreon 的 Web 界面，选择菜单 Configuration→Centreon→Pollers→Central→Monitoring Engine Information，确保 Monitoring Engine Binary 一项的值为 /usr/local/nagios/bin/nagios。

8.8.2 在 /var/log/messages 中出现 Warning: queue send error 错误

启动 ndo2db 服务后，如果在系统日志文件 /var/log/messages 中发现存在如图 8-30 所示错误：

```
Jun 10 13:12:48 localhost ndo2db-3x: Warning: Retrying message send. This can occur because you have too few messages allowed or too few total bytes allowed in message queues. You are currently using 64 of 1985 messages and 65536 of 65536 bytes in the queue. See README for kernel tuning options.
Jun 10 13:12:49 localhost ndo2db-3x: Message sent to queue.
Jun 10 13:12:49 localhost ndo2db-3x: Warning: queue send error, retrying...
Jun 10 13:12:50 localhost ndo2db-3x: Message sent to queue.
Jun 10 13:12:50 localhost ndo2db-3x: Warning: queue send error, retrying...
Jun 10 13:12:51 localhost ndo2db-3x: Message sent to queue.
Jun 10 13:12:51 localhost ndo2db-3x: Warning: queue send error, retrying...
Jun 10 13:12:52 localhost ndo2db-3x: Message sent to queue.
```

root@monitor:~/桌面 root@monitor:/etc/init.d

图 8-30 ndo2db 发送队列消息错误日志

即出现如下中的错误信息时，需要通过修改操作系统内核参数的方式解决此问题，方法如下：

```
"Jun 10 08:31:13 localhost ndo2db-3x: Warning: Retrying message send. This can occur because you have too few messages allowed or too few total bytes allowed in message queues. You are currently using 64 of 1985 messages
```

```
and 65536 of 65536 bytes in the queue. See README for kernel tuning options.
Jun 10 08:31:14 localhost ndo2db-3x: Message sent to queue.
Jun 10 08:31:14 localhost ndo2db-3x: Warning: queue send error, retrying...
Jun 10 08:31:15 localhost ndo2db-3x: Message sent to queue."
```

使用 `vi /etc/sysctl.conf` 命令修改系统配置文件, 适当扩大 `kernel.msgmnb` 和 `kernel.msgmax` 两项的值, 例如增大到 65536 的 8 倍。如果后续运行过程中仍出现该错误, 可以再适当增大这两个值。

```
# Controls the default maximum size of a message queue
kernel.msgmnb = 524288

# Controls the maximum size of a message, in bytes
kernel.msgmax = 524288
```

- 使用 `sysctl -p` 命令使配置生效。
- 使用 `service ndo2db restart` 命令重启 `ndo2db` 进程。

第9章

Centreon 的管理

从最开始的 Nagios 配置文件管理工具,到现在包罗万象的企业级 IT 运维监控和管理系统, Centreon 的演化令人印象深刻。在本章接下来的内容中,我们将探讨 Centreon 管理第三方调度进程和代理进程的原理、方式,以及典型的对于 Nagios 调度进程和 NDOUtils 代理进程的管理方式等。

9.1 Centreon 的调度进程和代理进程

如 7.5.4 小节和 7.5.6 小节中所述，调度进程（通常是 Nagios）和代理进程（通常是 NDOUtils）并没有包含在 Centreon 的软件发布包中。因为从 Centreon 的角度来讲，它们都是可以管理并替换的外部组件，只需要遵循下列 3 个步骤：

- （1）管理员通过 Centreon 的 Web 用户界面，对调度进程和代理进程的各项参数进行配置管理。这些参数存储在后台的 Centreon 数据库中。
- （2）通过 Centreon 的 Web 用户界面，管理员配置并生成调度进程和代理进程的各类文本格式配置文件。
- （3）Centreon 可以将生成的配置文件移动到合适的目录，并重启调度进程和代理进程以载入更改后的配置文件，使新的配置生效。

当然，以上步骤都可以绕开 Centreon 的 Web 用户界面，通过直接修改配置文件，并直接重启后台调度进程和代理进程的方式进行。但缺少了 Centreon 所提供的直观友好的 Web 界面，管理企业级的、庞大而复杂的监控项配置和运行参数配置就成了一项繁重到几乎不可能完成的任务。

9.2 Centreon 对于 Nagios 调度进程的管理

Nagios 进程的配置文件是 nagios.cfg。因此，Centreon 中，对于 Nagios 调度进程的管理基本上是通过管理 nagios.cfg 文件来进行的。

首先，进入如图 9-1 所示的 Centreon 菜单 Configuration→Monitoring Engine→Nagios。

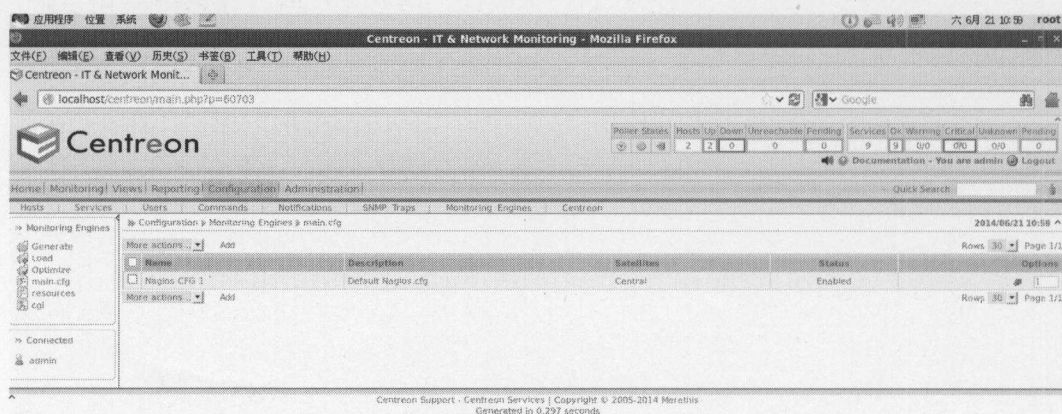


图 9-1 Centreon 的 Nagios 配置文件项

上图 9-1 中显示了 Centreon 中配置的所有 Nagios 调度进程列表项，Centreon 支持管理多个 Nagios 调度进程，因此在列表项中可存在超过 1 项 Nagios 配置文件信息。在列表配置项中，Name、Description 两项均可由管理员自由设置名称，而 Satellites 项则标识了调度进程是属于默认的中央 Nagios 调度器进程，还是分布式的 Nagios 调度进程。由于我们之前已经在服务器上同时部署了 Nagios 和 Centreon 服务，因此默认的 Nagios CFG 1 配置项便是默认的中央调度进程。如果想为其他分布式的 Nagios 调度进程创建配置文件，只需要选中默认的配

置项，在下拉列表中选择 Duplicate 选项即可复制出一份新的配置项，并在复制配置项的基础上修改即可，如图 9-2 所示。

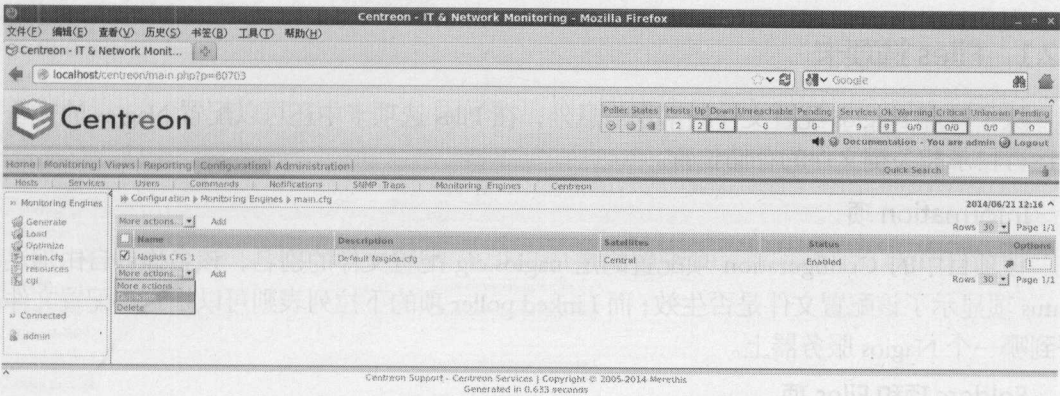


图 9-2 在 Centreon 中复制 Nagios 配置项

复制出的配置项中，默认的 Status 项是 Disabled 状态，标明该项并未生效，单击 Options 中的箭头图标则会设置该项为 Enabled 状态，说明该配置项已经被启用。

注意：在实际的应用部署中，这种利用 Centreon 管理分布式调度进程的方式在管理上各有利弊。其优点是可以利用 Centreon 管理多个版本的调度进程，例如不同版本的 Nagios，或者 Centreon 所提供的 Centreon Engine 等，体验并总结每种调度进程的优势。而缺点在于需要配置多个不同的配置项，带来管理上的不便甚至混乱。因此，在实际项目实施中，请尽量保持调度引擎版本的一致，有利于后续的配置管理等。

接下来，单击菜单中默认的 Nagios CFG 1 项，进入信息编辑页面，如图 9-3 所示。

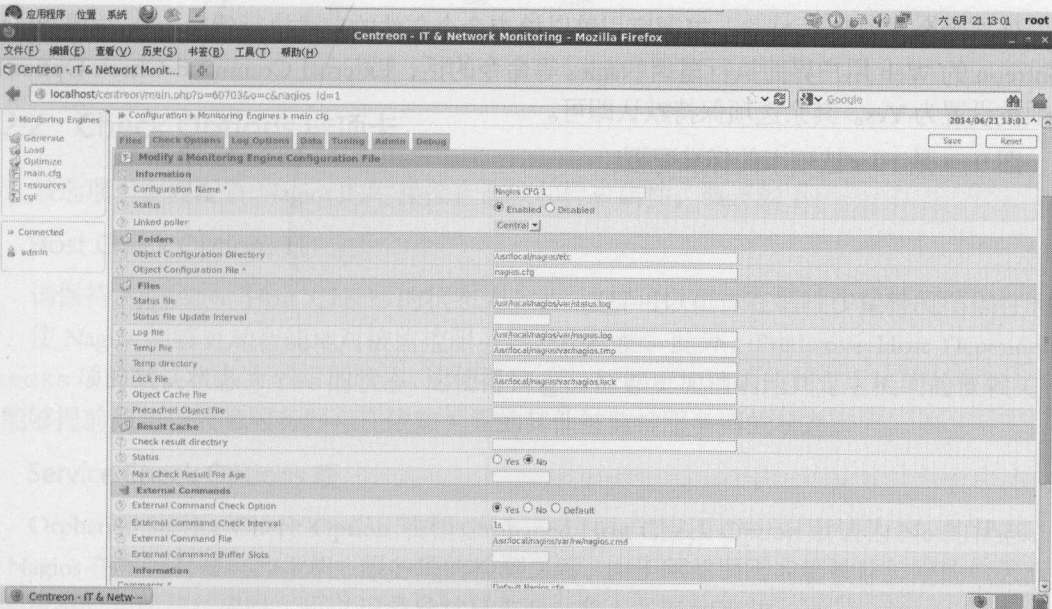


图 9-3 Nagios 配置文件信息编辑页面

如上图 9-3 所示，对于 Nagios 的管理是通过配置 7 类选项卡进行的，单击每一个行首的 ? 符号，可以找到该配置项的相关说明，下面就对这 7 类配置项进行详细介绍。

9.2.1 Files 选项卡

除了管理 Nagios 配置文件的常用信息外，在 Files 选项卡中还可以配置 Nagios 的一些关键运行目录和关键文件的存储位置。

Information 项

该项目中的 Configuration 项配置的是 nagios.cfg 配置文件的别名，该名称可自由指定；Status 项显示了该配置文件是否生效；而 Linked poller 项的下拉列表则可以指定该配置文件部署到哪一个 Nagios 服务器上。

Folders 项和 Files 项

这两项用来管理 Nagios 关键配置文件的服务器存放路径和文件名，并据以在 Centreon 中产生相关配置，便于后者对于 Nagios 的管理。这些存放路径都是基于 Nagios 的默认标准，一般无需更改。

Result Cache 项

该项是对 Nagios 等调度引擎存放临时文件的方式进行管理和配置。Nagios 会将检查结果临时存放在 Check result directory 指定的服务器路径里，待后续处理所用。需要注意的是一旦配置了多个 Nagios 调度引擎，该项的 Status 应该设为无效，否则多个引擎争用该临时目录里的临时检测文件，将会造成不可预见的后果。

External Commands 项

外部命令选项允许 Nagios 等调度引擎以检查命令文件的方式执行外部命令，如果想通过 Centreon 的 Web 用户界面执行重启 Nagios 等命令的话，External Command Check Option 项必须被设置为 Yes。其余选项保持默认即可。

图 9-4 是 Files 选项卡的标准配置。

Configuration > Monitoring Engines > main.cfg

2014/06/22 9:53

Files | Check Options | Log Options | Data | Tuning | Admin | Debug

Save | Reset

Modify a Monitoring Engine Configuration File

Information

Configuration Name *
Nagios CFG 1

Status
☒ Enabled ☐ Disabled

Linked poller
Central

Folders

Object Configuration Directory
/usr/local/nagios/etc

Object Configuration File *
nagios.cfg

Files

Status file
/usr/local/nagios/var/status.log

Status file Update Interval

Log file
/usr/local/nagios/var/nagios.log

Temp file
/usr/local/nagios/var/nagios.tmp

Temp directory

Lock file
/usr/local/nagios/var/nagios.lock

Object Cache File

Precached Object File

Result Cache

Check result directory

Status
☐ Yes ☒ No

Max Check Result File Age

External Commands

External Command Check Option
☒ Yes ☐ No ☐ Default

External Command Check Interval
1s

External Command File
/usr/local/nagios/var/hw/nagios.cmd

External Command Buffer Slots

Information

Comments *
Default Nagios.cfg

☒ List ☐ Form

Save | Reset

图 9-4 Files 选项卡的标准配置

9.2.2 Check Options 选项卡

该选项卡主要配置 Nagios 的附件检查选项和检查属性，概述如下：

Host Check Options 项

请保持 Aggressive Host Checks 的状态为 No，这样可以确保在牺牲少量检测精确性的同时，使 Nagios 能够更加智能地对被监控服务器发起检测；此外，Predictive Host Dependency Checks 项的默认状态为 Yes 的状态，这使得 Nagios 能够更加精确地判定主机间的依赖关系，并能够提前预判与故障服务器存在依赖关系的其他被监控服务器的状态。

Service Check Options 项

Orphaned Service Check Option 项和 Orphaned Host Check Option 默认为 No 的状态，这样 Nagios 在发起对被监控主机、服务项的检测之后，由于网络原因或者被监控端的原因而长时间没有收到检测结果时，可以记录异常日志而非陷入长时间等待。

Soft Service Dependencies Option 选项默认为 No 状态，可以使 Nagios 在遇到监控项或者主机处于硬状态(参考 9.5.3 小节)后，才会判定相关存在依赖关系的监控项或者主机的状态，

而非监控项每产生一次状态变化都要通知到相关监控项,如此一来可以避免频繁的无效告警。

Predictive Service Dependency Checks 选项默认为 Yes 状态,可以使 Nagios 在遇到监控项故障后,能预先对相关存在依赖关系的监控项的状态做出判断。

Event Handler 项

在遇到被监控主机或者监控项产生故障告警、或者从故障中恢复等状态变化时, Nagios 提供了一种能够执行 Global Host Event Handler 或者 Global Service Event Handler 全局脚本的机制。用户可以预先定义全局的主机事件处理脚本或者监控项事件处理脚本,例如发送一段邮件、短信通知到相应系统管理员等等。除了全局脚本之外,还可以为每个被监控主机或者监控项单独指定事件处理脚本,这在故障通知、故障紧急处理、故障应急恢复,与其他监控系统集成等方面非常有用。

Freshness 项

该项在被动检测机制下有用,可以指定全局的检测结果“新鲜”程度,确保在被动模式下, Nagios 能够周期性地收到被监控项的检测结果,否则便会发起一次主动检测。在 Service Freshness Check Interval 中和 Host Freshness Check Interval 指定在一天(86400 秒)之内期望多少秒收到一次检测结果。

Flapping Options 项

当被监控主机或者监控项的状态变化变得频繁时, Nagios 称此类现象为“抖动”,并提供相应的抖动检测机制,以避免告警过频的问题。建议设置 Flap Detection Option 项为 Yes 状态,并保持其他项目为默认状态。

Post Check 项

在 Nagios 分布式架构中,位于远程的被监控项每次执行完检测,都要回传检测结果到中央服务器上,这就是被动监控的基本行为模式。在该项中,通过设置 Obsess Over Services Option 为 Yes 的状态,可以启用 Nagios 的分布式被动检测机制。由于本书中采用 Nagios 主动检测机制,因此该项设置为 No。

Obsessive Compulsive Service Processor Command 项中可以定义每次执行完主机或者监控项检查后执行的命令,例如将数据传到监控中心(submit_host_check_result)等等。

Misc Options 项

该项目下的各类选项涉及到了 Nagios 最基本的特性,且为全局特性,保持默认即可。在 Nagios 的设计中,单台被监控主机或者监控项同样具备此类属性,且可以修改。在日常管理中,我们可以单个或者批量为监控项设置这些属性,其优先级高于全局属性。

Passive host checking Options 项

该项目下的配置依然与 Nagios 被动检测机制有关,保持默认即可。图 9-5 是 Checks Options 选项卡的标准配置。

Configuration » Monitoring Engines » main.cfg2014/06/22 9:09

FileCheck OptionsLog OptionsDataTuningAdminDebug

Modify a Monitoring Engine Configuration File

Host Check Options

Aggressive Host Checks

☒ No

This option must be disable in order to avoid latency problem.

Predictive Host Dependency Checks

☐ Yes ☐ No ☒ Default

Service Check Options

Orphaned Service Check Option

☐ Yes ☒ No ☐ Default

Orphaned Host Check Option

☐ Yes ☒ No ☐ Default

Soft Service Dependencies Option

☐ Yes ☒ No ☐ Default

Predictive Service Dependency Checks

☐ Yes ☐ No ☒ Default

Event Handler

Global Host Event Handler

Global Service Event Handler

Freshness

Service Freshness Check Option

☒ Yes ☐ No ☐ Default

Service Freshness Check Interval

 seconds

Host Freshness Check Option

☐ Yes ☐ No ☒ Default

Host Freshness Check Interval

 seconds

Additional freshness latency

 seconds

Flapping Options

Flap Detection Option

☐ Yes ☒ No ☐ Default

Low Service Flap Threshold

25.0

 %

High Service Flap Threshold

50.0

 %

Low Host Flap Threshold

25.0

 %

High Host Flap Threshold

50.0

 %

Post Check

Obsess Over Services Option

☐ Yes ☒ No ☐ Default

Obsessive Compulsive Service Processor Command

Obsess Over Hosts Option

☐ Yes ☐ No ☒ Default

Obsessive Compulsive Host Processor Command

Misc Options

Notification Option

☒ Yes ☐ No ☐ Default

Service Check Execution Option

☒ Yes ☐ No ☐ Default

Passive Service Check Acceptance Option

☒ Yes ☐ No ☐ Default

Event Handler Option

☒ Yes ☐ No ☐ Default

Host Check Execution Option

☐ Yes ☐ No ☒ Default

Passive Host Check Acceptance Option

☐ Yes ☐ No ☒ Default

Passive host checking Options

Translate Passive Host Checks Option

☐ Yes ☐ No ☐ Default

Passive Host Checks Are SOFT Option

☐ Yes ☐ No ☐ Default

☒ List ☐ Form

SaveReset

图 9-5 Check Options 选项卡的标准配置

9.2.3 Log Options 选项卡

该选项卡主要用来配置 Centreon 系统的运行、告警类日志信息，以及各类运行图表、报告等信息。通过配置该选项卡里的各类属性，您可以随意调整数据库中存储的日志文件的大小、有效时间等重要配置项，避免用户在使用 Centreon 的过程中，因存储日志过多导致的用户界面响应时间过长等问题。

Logging Options 项

该组选项允许您控制 Nagios 记录的各类日志的类型和大小, 并因此而影响到 Centreon 所能够显示的日志内容等。在默认状态中, 我们需要注意 Service Check Retry Logging Option 项和 Host Retry Logging Option 项, 启用这两项会使 Nagios 在被监控对象的状态频繁变化 (软状态) 时也会记录相关日志, 导致日志性能的下降。因此后期对于 Nagios 日志性能有特别关注的管理员可以禁用这两项。

默认情况下, Nagios 只记录告警信息和恢复信息, 而并不记录首次检测的状态正常信息, 这会造成监控项状态检测信息的不一致。为了使 Centreon 能够统计并显示完整的监控项检测信息和相关报表, Initial State Logging Option 项默认为 Yes 状态。

Timeouts 项

该组选项允许您定义主机检测和服务检测的超时时间。一般来说, 服务检测在 60 秒以内, 主机检测应该在 10 秒之内, 如果超过各自的时限仍未上报检测信息, 相关的检测进程就会终止, 相应的检测状态就会标记为 Unknown, 而被监控项上报给 Nagios 的检测结果就会是 “检测超时” 或者 “连接超时”。需要注意的是, 长时间的监控结果等待意味着 Nagios 性能的损耗, 在关注调度进程性能的管理员眼中, 这是个尤其需要重点关注的问题。

Archives 项

该组选项用来设置 Nagios 对于检测日志的归档周期和归档服务器位置。一般来说, Nagios 运行过程中产生的检测日志没有多大用处, 但是解析后可以用来重新填充数据库中相关表格里的记录。此处建议保留 30 天的日志信息。需要注意的是, Centreon 并未提供自动清除日志的机制, 需要通过运行操作系统 cron 定时任务驱动脚本的方式来定期清除 Nagios 的检测日志, 相关清除脚本位于 Centreon 的安装目录 /usr/local/centreon/cron 下。

States Retention 项

该组选项允许 Nagios 在服务器文件中缓存监控项信息和状态信息, 避免每一次重启后都需要重新读取数据库、重新生成各类配置信息, 这会严重影响检测性能。而利用 Nagios 的缓存机制, Centreon 也能够第一时间显示 Nagios 缓存的各类监控项和状态信息。

如果 State Retention Option 项设置为 Yes, Nagios 就会定期将各类信息缓存至 State Retention File 指定的服务器路径中。

以上缓存文件在 Nagios 的每一次启动后都会加载, 并实时显示在 Centreon 的 Web 用户界面中。启用 Use Retained Program State Option 为 Yes 状态可使 Nagios 从缓存中读取一些全局的配置信息, 加快 Nagios 的启动速度。而设置 Use Retained Scheduling Info Option 项为 Yes 状态可以缓存监控项各自的检测时间, 当 Nagios 重启之后, 会根据缓存的检测时间调度相应的检测项, 而非重新发起检测。一般情况下, 这两项默认均为 Yes 状态。

Log Options 选项卡的默认配置如图 9-6 所示。

File

Check Options

Log Options

Data

Tuning

Admin

Debug

Save

Reset

Modify a Monitoring Engine Configuration File

Logging Options

?

 Syslog Logging Option

☐ Yes

☒ No

☐ Default

?

 Notification Logging Option

☒ Yes

☐ No

☐ Default

?

 Service Check Retry Logging Option

☒ Yes

☐ No

☐ Default

?

 Host Retry Logging Option

☒ Yes

☐ No

☐ Default

?

 Event Handler Logging Option

☒ Yes

☐ No

☐ Default

?

 Initial State Logging Option

☒ Yes

☐ No

☐ Default

This option must be enabled for Centreon Dashboard module.

?

 External Command Logging Option

☒ Yes

☐ No

☐ Default

?

 Passive Check Logging Option

☐ Yes

☐ No

☒ Default

Timeouts

?

 Service Check Timeout

80

seconds

?

 Host Check Timeout

10

seconds

?

 Event Handler Timeout

30

seconds

?

 Notification Timeout

30

seconds

?

 Performance Data Processor Command Timeout

5

seconds

?

 Obsessive Compulsive Service Processor Timeout

5

seconds

?

 Obsessive Compulsive Host Processor Timeout

5

seconds

Archives

?

 Log Rotation Method

☐ None

☐ Hourly

☒ Daily

☐ Weekly

☐ Monthly

?

 Log Archive Path

/usr/local/nagios/var/archives/

States Retention

?

 State Retention Option

☒ Yes

☐ No

☐ Default

?

 State Retention File

/usr/local/nagios/var/retention.dat

?

 Automatic State Retention Update Interval

60

minutes

?

 Use Retained Program State Option

☒ Yes

☐ No

☐ Default

?

 Use Retained Scheduling Info Option

☒ Yes

☐ No

☐ Default

?

 Retained Contact Host Attribute Mask

?

 Retained Contact Service Attribute Mask

?

 Retained Process Host Attribute Mask

?

 Retained Process Service Attribute Mask

?

 Retained Host Attribute Mask

?

 Retained Service Attribute Mask

☒ List

☐ Form

Save

Reset

图 9-6 Log Options 选项卡的默认配置

9.2.4 Data 选项卡

该选项卡主要是用来配置并管理 Centreon 对于监控数据的处理方式。Nagios 提供了多种手段用以备份并处理采集到的监控数据，而 Centreon 使用了其中的两种：

- 使用 ndomod 模块实时存储监控数据。
- 调用相应进程处理监控数据，并显示监控项的性能信息。

Modify a Monitoring Engine Configuration File 项

该组选项允许用户添加多个 Nagios 监控数据代理进程（Broker），并设置这些进程的启动命令。注意 NDO 组件的启动选项始终是 -1，确保 Nagios 能够将所有的监控数据交给 ndo2db 组件来处理。

Perfdata 项

Nagios 可提供多种手段备份并处理监控数据。而 Centreon 仅使用其中一种方式，即调用

process-service-perfdata 进程来处理监控数据，并将处理过的监控数据存放在服务器目录 /usr/local/nagios/var/service-perfdata 目录下。

图 9-7 是 Data 选项卡的典型配置。

The screenshot shows the 'Data' tab in the Nagios configuration interface. The page title is 'Modify a Monitoring Engine Configuration File'. The breadcrumb trail is 'Configuration > Monitoring Engines > main.cfg'. The date and time are '2014/06/23 16:57'. There are 'Save' and 'Reset' buttons at the top right. The main content area is divided into sections: 'Broker Module', 'Perfdata', and 'Host Performance Data File Mode'. The 'Broker Module' section has a 'Multiple Broker Module' note and an 'Event broker directive' field. The 'Perfdata' section has a 'Performance Data Processing Option' dropdown set to 'Yes', and several other options for host and service performance data processing commands and file templates. The 'Host Performance Data File Mode' section has radio buttons for 'Append', 'Write', and 'Default', with 'Default' selected. There are also input fields for 'Host Performance Data File Processing Interval' and 'Service Performance Data File Processing Interval'. At the bottom, there are 'List' and 'Form' radio buttons, and 'Save' and 'Reset' buttons.

图 9-7 Data 选项卡的典型配置

9.2.5 Tuning 选项卡

该选项卡主要涉及到 Nagios 的性能优化，一般来说保持初始设置默认不变即可，详细的优化手段在后续章节会讲到，以下是各组选项的简要介绍。

Time Unit 项

选项 Inter-Check Sleep Time 的默认值是 1，标明了 Nagios 在两次主动检查之间需要等待的时间，单位是秒。注意 Nagios 对于检测项的检查调度是基于时序的队列方式实现的，一旦 Nagios 发现对于某项监控项的检测时间并非按照预先定义的时序进行，会自动等待一定时间，使实际的检测时间与预先定义的检测时序队列同步。

Host Check Scheduling Options 项和 Service Check Scheduling Options 项

Nagios 调度引擎的工作机制设计得非常巧妙，所用到的 Host Inter-Check Delay Method 选项、Service Inter-Check Delay Method 选项，以及 Service Interleave Factor 选项均为 s，使 Nagios 能够以 Smart（智能）方式实施主机和服务检查智能调度策略，以大幅降低服务器负

面的属性。

在 Monitoring system User 和 Monitoring system Group 项中, 指定的是 Nagios 调度引擎的用户和属组, 毫无疑问应该是 nagios, 如果您在安装过程中指定了其他用户, 那么填入相应的用户名和属组值即可。而后面的一些涉及到日期格式及非法字符的选项, 可以保持默认值。而 Administrator Email Address 和 Administrator Pager 指的是系统管理员的邮箱地址和寻呼机、手机等号码, 以便后续告警时发送通知消息所用, 根据实际情况填入相应值即可。

剩下的 Perl 选项组与 Nagios 内嵌的 Perl 语言解释器有关，保持默认即可。

图 9-9 是 Admin 选项卡的默认配置。

Configuration > Monitoring Engines > main.cfg

Files Check Options Log Options Data Tuning Admin Debug

Save

Modify a Monitoring Engine Configuration File

Monitoring system User: nagios

Monitoring system Group: nagios

Date Format: euro (30/06/2002 03:15:00)

Illegal Object Name Characters: ~!\$%^&*~!<>?.,()=

Illegal Macro Output Characters: ~\$&*~!<>

Regular Expression Matching Option: ☐ Yes ☐ No ☒ Default

True Regular Expression Matching Option: ☐ Yes ☐ No ☒ Default

Administrator Email Address: admin@localhost

Administrator Pager: admin

Perl

Enable embedded Perl: ☐ Yes ☐ No ☒ Default

Use embedded Perl implicitly: ☐ Yes ☐ No ☒ Default

Embedded Perl initialisation file: /usr/local/nagios/bin/p1.pl

☒ List ☐ Form

Save Reset

图 9-9 Admin 选项卡的默认配置

9.2.7 Debug 选项卡

该选项卡允许您为 Nagios 指定一个 Debug 文件，并可以选择写入不同类型的 Debug 信息。在实际运行过程中，往往没有必要指定该文件，保持参数默认即可，如图 9-10 所示。

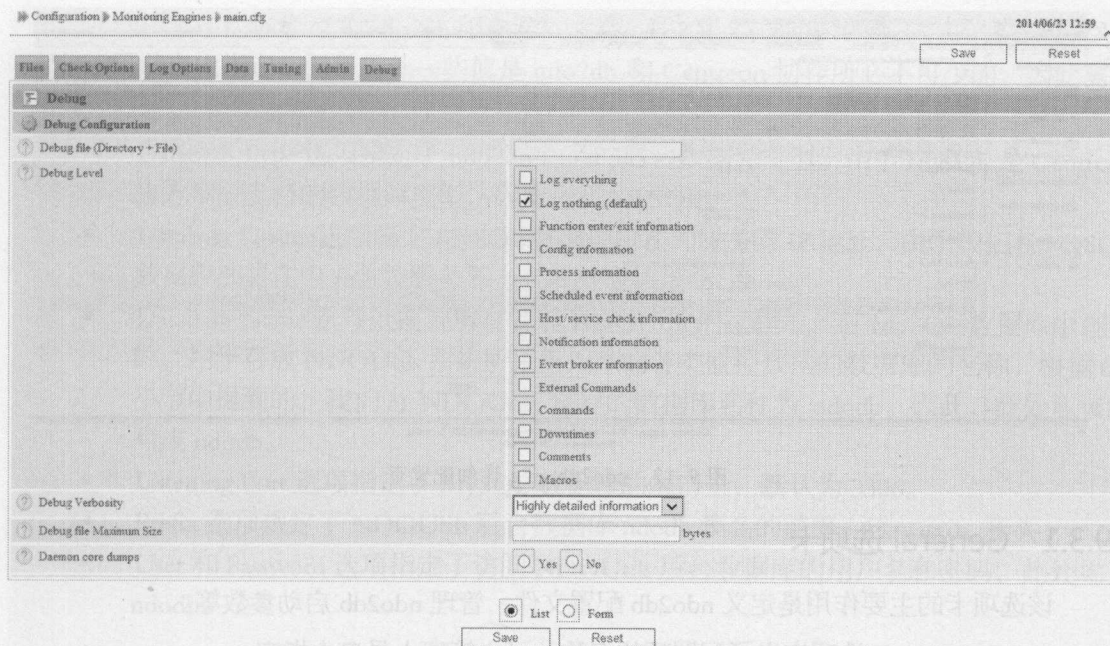


图 9-10 Debug 选项卡的标准配置

9.3 Centreon 对于 NDOUtils 代理进程的管理

与管理 Nagios 调度进程的方式类似，Centreon 对于 NDOUtils 的管理也是在 Web 用户界面中，通过对于 NDOUtils 的配置文件—ndo2db.cfg 的管理来实现的。单击菜单项 Configuration—Centreon，在左侧的竖状菜单里，单击 ndo2db.cfg 菜单项，如图 9-11 所示。

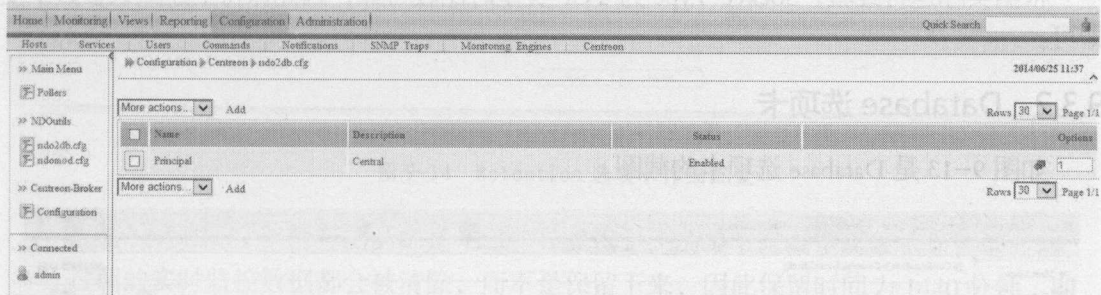


图 9-11 ndo2db 菜单项

在典型的 Centreon&Nagios 集成架构中，ndo2db 作为单一的代理组件是必不可少的，它一般位于 Centreon 中央监控服务器上。

接下来，单击图 9-11 中的 Principal 项，进入 ndo2db.cfg 的配置页面，有三个选项卡，如图 9-12 所示。

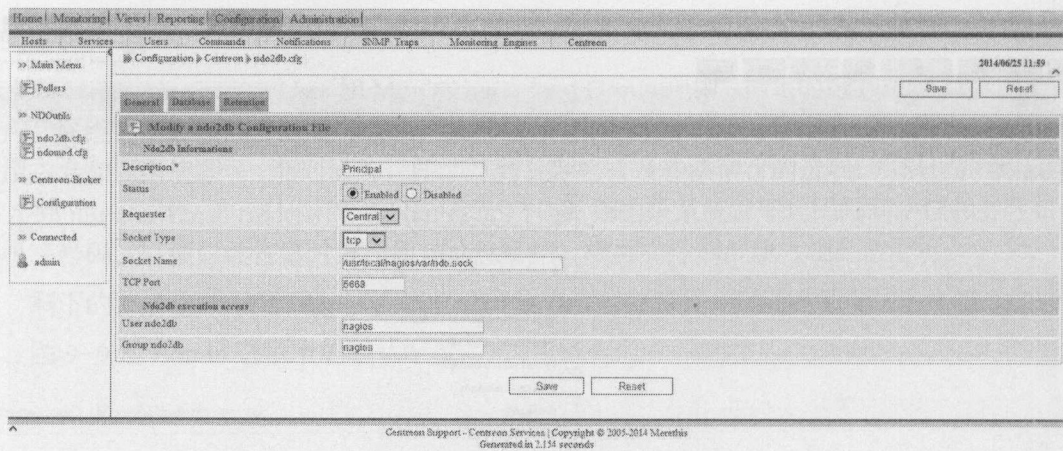


图 9-12 ndo2db.cfg 详细配置页

9.3.1 General 选项卡

该选项卡的主要作用是定义 ndo2db 配置文件，管理 ndo2db 启动参数等。

- Description 选项定义了配置项的名称，可由管理人员自由指定。
- Status 选项定义了配置文件是否生效。
- Requester 选项指定了监控数据的存放位置，通常都是中央监控服务器。
- Socket Type 选项指定了 ndo2db 进程使用的端口类型，一般都是 TCP。
- Socket Name 选项定义了 UNIX 套接字文件的路径。
- TCP Port 选项定义了 ndo2db 组件运行时所监听的端口号。

根据实际运行经验，Socket Type 为 TCP 类型时比较可靠，因此保持上述默认参数不变即可。

9.3.2 Database 选项卡

如图 9-13 是 Database 选项卡的截图。

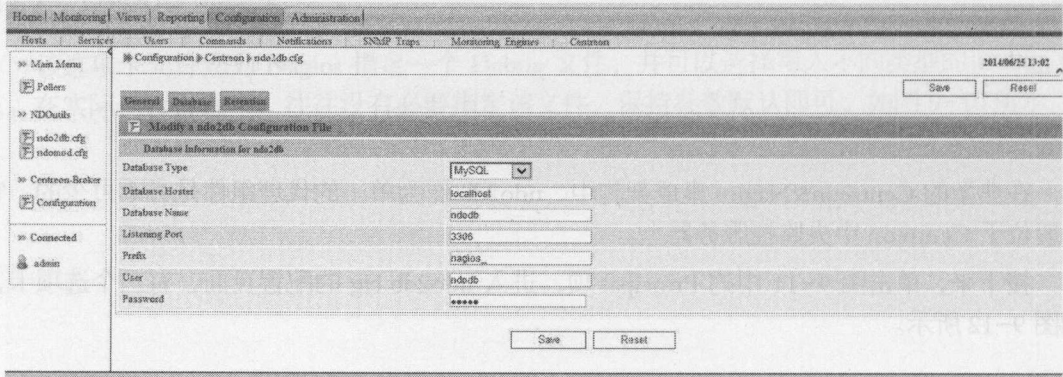


图 9-13 Database 详细配置页

此选项卡中允许您定义 ndo2db 组件访问后台 MySQL 数据库的一些连接设置参数。尽管有些值可以自由指定，但是另外一些值是 ndo2db 和 Centreon 协作所必不可少的，因此需要注意。

- Database Type 选项指定了 ndo2db 访问的后台数据库类型。由于 MySQL 是 Centreon 官方唯一支持的数据库类型，因此选择 MySQL。
- Database Host 选项指定 MySQL 数据库所在的服务器 IP 地址，由于我们将 MySQL 数据库部署在中央监控服务器上，此处应该是 localhost。
- Database Name 选项处需要填写安装 NDOUtils 的过程中，在 MySQL 数据库中创建的，用于存放 NDOUtils 系统所采集入库的各类监控数据的数据库的名称，根据 6.4 小节中提到的，我们为 NDOUtils 创建的数据库名称为 ndodb。因此，此处应该填写成 ndodb。
- Listening Port 选项指定了 MySQL 数据库的端口号，默认为 3306。
- Prefix 选项规定了 NDOUtils 后台数据库 ndodb 表名的前缀，以 nagios_ 开头。
- User 和 Password 选项指定了访问 NDOUtils 后台数据库的用户名和密码，此处均为 ndodb。

9.3.3 Retention 选项卡

如图 9-14 所示，是 Retention 选项卡的界面。

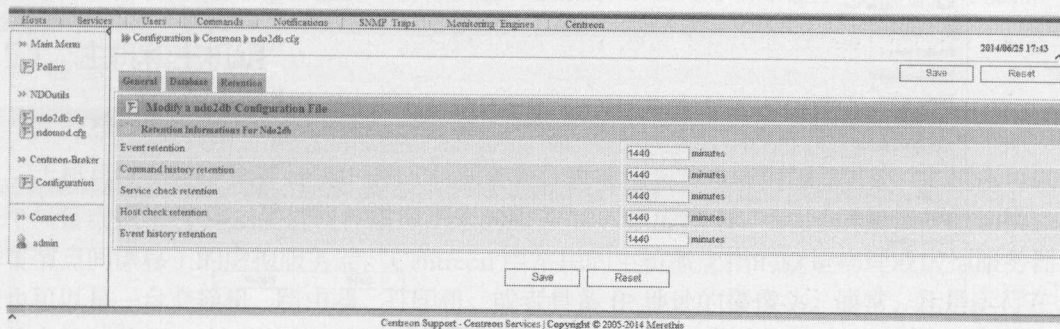


图 9-14 Retention 选项卡详细配置

注意：NDOUtils 组件存储的都是实时监控数据，此选项卡可以定义数据保留的时间。过期的实时监控数据都会被清除，而不是保留下来，因此保留时间为 1440 分钟，即保留 1 天的数据即可。

9.4 Centreon 对于 ndomod 的管理

如图 9-11，可以通过单击左侧竖状菜单里的 ndomod.cfg 选项，进入 ndomod.cfg 配置文件的列表，如图 9-15 所示。

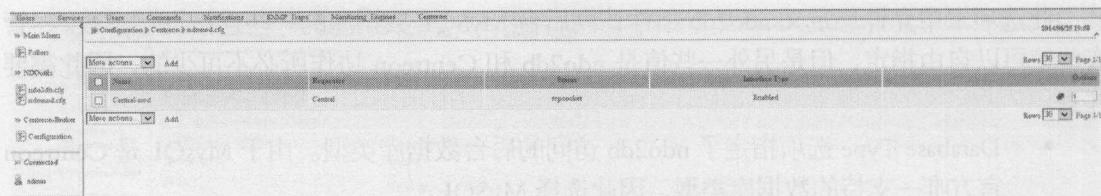


图 9-15 ndomod 配置文件列表

单击上图 9-15 中的 Central-mod, 进入 Centreon 对于 nodmod.cfg 文件的配置界面, 如图 9-16 所示。

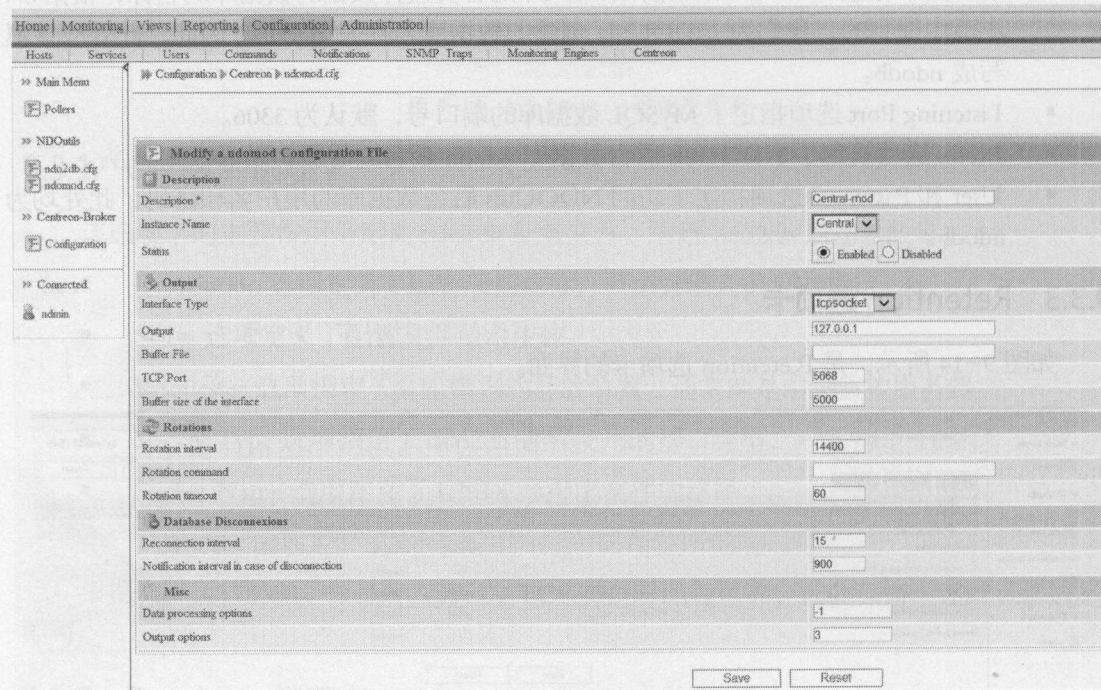


图 9-16 ndomod 配置界面

- Description 选项指定了 ndomod 配置项的名称, 可由用户自定义。
- Instance Name 选项定义 ndomod 部署的位置, 一般是位于中央监控服务器上。
- Status 选项定义配置项是否生效, 该项始终为 Enabled 状态。

接下来的 Output 选项组用来定义 ndomod 连接 ndo2db 的属性。

- Interface Type 选项, 应该与图 9-12 中 ndo2db 定义的 Socket Type 选项一致, 此处为 tcpsocket。
- Output 选项, 指的是 ndo2db 部署并运行的服务器地址, 由于我们将 ndo2db 和 ndomod 运行在中央监控服务器上, 此处应该是 127.0.0.1。
- Buffer file 选项, 允许指定一个临时缓存文件, 当 ndomod 失去与 ndo2db 连接后缓存临时数据用。

- TCP Port 选项，定义了 ndomod 与 ndo2db 连接所用的端口号，即 5668。
- Buffer size of the interface 选项，指定了 Buffer file 选项定义的临时缓存文件中允许存放的最大消息数量。

Rotations 选项组允许您指定一个可以旋转的缓存控制文件，该选项只有在设置 Interface Type 选项为 file 类型时才有效，因此保持该选项组为默认设置。

Database Disconnexions 选项组设置当 ndomod 和 ndo2db 组件与后台数据库失去连接后，重新连接数据库的相关参数。Reconnection interval 设置重新连接的间隔时间，Notification interval in case of disconnection 设置从连接断开到发出第一条通知消息之间的间隔时间。该组选项保持默认即可。

在接下来的杂项中，Data processing options 默认值为-1，它与 Centreon 处理事件的类型有关；而 Output options 为 3，以确保 Centreon 能够正确处理 ndomod 和 ndo2db 组件采集的数据，该两个选项保持默认值。

9.5 Centreon 的实时监控

在 Centreon 中，每一个被监控的主机和被监控项都是监控对象，Centreon 负责判断这些监控对象的可用性，即对外提供服务的能力，以及监控对象的性能数据，例如服务器 CPU 利用率等因素，并以此判定这些对象的状态，这就是 Centreon 对于监控对象的实时监控。

9.5.1 主机和主机组

1. 主机

主机可以是一个具有 IP 地址的实体服务器，也被称为节点或资源。另外，主机还可以是一个具备虚拟 IP（即由集群软件虚拟出来的服务 IP，该虚拟 IP 可以随群集资源组切换，在群集节点间漂移）的虚拟服务器。Centreon 中常用的主机概念指的就是物理或虚拟服务器，但也可以是一台交换机、路由器、打印机、或者具备 IP 地址的摄像头。通常，凡是运行在 IP 网络内的，具备 IP 地址的物理设备或者虚拟设备，都可以配置为在 Centreon 的主机。

Centreon 控制主机的可用性。一台主机的状态可以有如下 4 类状态：

- UP：主机可用。
- DOWN：该主机不可用，即主机处于宕机状态。
- 不可达：主机不可达。通常是由于该主机所依赖的上游设备（通常是一个交换机或路由器）的不可用。
- 未知：主机的状态是未知的，通常是因为外部原因导致的状态无法验证（检测项出现错误，SNMP 禁用或配置项错误等）。

一般来说，Centreon 都是通过一个简单的 ping 命令检查主机是否可用，然而 Centreon 允许自由使用命令来检查主机的可用性，例如，可以通过访问一个网页来监控 Web 服务器的可用性，这通常比使用简单的 Ping 命令检查主机是否在线存活要有意义的多。

2. 主机组

Centreon 支持将一个或多个主机配置到主机组中。这些主机组群体的命名可能有不同的语义：基于平台或者技术（如 Linux 服务器）、基于地理位置（如中央商务区，省份等等地理位置）、基于应用程序、或基于业务（如人力资源信息系统、ERP 系统等）。对于主机组的操作通常集中于权限控制，通知升级，或者简单地分类或者过滤。

9.5.2 服务、服务组和元服务

1. 服务

服务是位于被监控主机上的业务控制点。一般来说，服务从属于主机，不论是物理主机还是虚拟主机，即服务不能脱离主机而单独存在。一个服务通常是指一项关于主机某个指标的度量或指示。服务可以是在信息系统层级结构中的任何一层，例如物理层、系统软件层、应用程序层、业务或进程层面等等。

以下是服务的一些示例：一个 Ping 命令的延迟、磁盘空间使用率、CPU 使用率、机房温湿度、风扇转速、数据库可用性、网页访问延迟、订单状态等。

Centreon 能够监控服务的性能。要做到这一点，需要基于两个阈值（警告和严重），使服务的性能能够被评估。以下是服务可能的 4 类状态：

- 正常：该服务运行正常，可正常提供服务。
- 警告：该服务被降效（实际提供的对外服务也许依然保持正常），但其典型表现是高于警告阈值但低于紧急临界阈值。
- 紧急：该服务需要立即进行干预，其监控值已经超过紧急状态临界值，属于危重状态。此时服务有可能依然正常，超过阈值并不意味着服务不可用，只是需要紧急干预，否则极有可能导致服务不可用。
- 未知：服务状态是未知的，因为外部原因导致服务暂时无法被验证（服务监控项执行失败，SNMP 禁用或配置项错误等）。

2. 服务组

与主机组的用法类似，服务组的目的是将多个服务以语义关联在一起，达到便于管理的目的。例如，可以将某些提供关键数据库服务的 Oracle 监控项组合成 Oracle 数据库系统服务组，以便于 Oracle 数据库系统的监控。

服务组可以在日常配置管理、通知上报配置等常用的管理方式中运用。在实践中，通常将相关服务聚合成服务组，以便于在业务性能展示、业务视图以及多维汇总分析中使用。

3. 元服务

元服务是指其性能数据是通过使用数学运算，例如求和、平均值、最大值或最小值等方式，聚合来自其他服务的数据而构建的服务。和正常的服务一样，元服务同样具备告警通知机制，并且具备性能曲线图。

元服务是 Centreon 独有的概念，在 Nagios 中不存在元服务的概念。例如，可以将一台服务器中多个网卡提供的，各自独立的网络服务带宽聚合成网络带宽元服务，从而显示出这

台服务器的网络聚合图形,运用同样的方式,可以计算出一组 Web 集群节点能够提供的网络连接数的总和或者平均值。

当定义元服务时,一定要注意各项服务指标单位的一致性,例如将多个网卡的网络服务带宽聚合成元服务时,带宽的单位一定要全部是字节、千字节或者是千兆字节,不能相互之间不一致,否则会导致无法叠加计算或取平均值。

9.5.3 硬状态和软状态

当监控项发生异常时(宕机、警告、紧急或未知),在已经配置监控项相关参数的前提下,Centreon 可以在执行消息通知之前,主动发起若干连续的检查,以确保监控项状态确实发生了改变。换句话说,在错误状态未经证实之前,Centreon 使用如下的状态来对监控项的“错误”状态进行进一步证实。

- **软状态:** 状态未证实,不会触发下一步的通知消息。
- **硬状态:** 条件确定后,会触发下一步的通知消息,例如相关联系人可以得到通知。

设立硬状态和软状态的目的是为了减少通知的数量,避免状态频繁变化造成的频繁通知。

相关配置

最大检查尝试次数,即确认状态变“硬状态”之前需要执行检查的次数,即图 9-17 中的 Max Check Attempts,每次检查之间的时间间隔,即 Normal Check Interval,以及重试间隔 Retry Check Interval 这 3 项参数,都可以在主机和服务的相关配置项中进行配置。

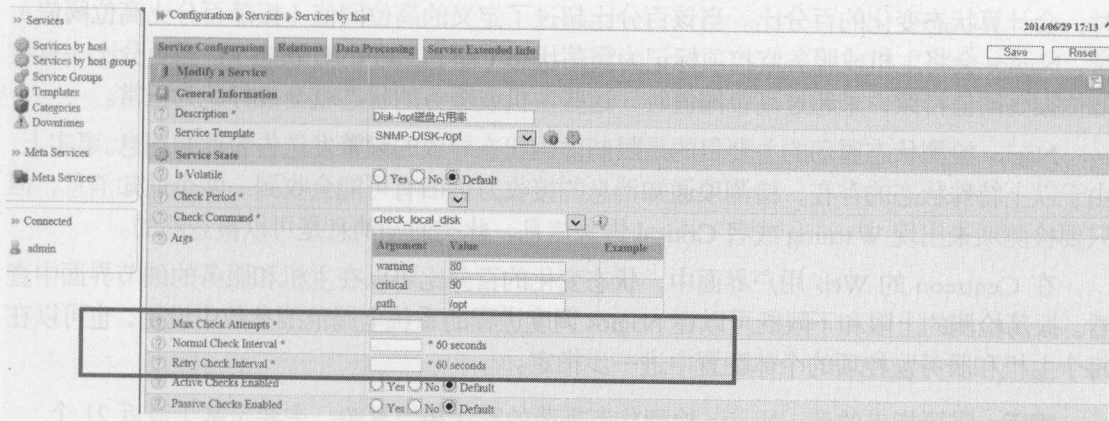


图 9-17 监控项的检查次数与检查间隔配置

如果设置 Max Check Attempts 大于 1,当 Centreon 检测到监控项的状态发生变化时,例如从 OK 状态到 Warning 状态,那么会保持变化后的状态为“软(SOFT)”状态,此时不会触发通知消息。Centreon 会按照重试间隔多次执行重试检测,直到达到最大重试次数为止。如果此时监控项的状态始终保持为变化后的状态,即 Warning 状态,Centreon 就会触发一条 Warning 通知消息。即一旦状态被证实,Centreon 会确认该监控项的状态由“软”状态转变为“硬(HARD)”状态。

图 9-18 显示出在监控项状态的变化下,Centreon 所执行的检测和通知动作的变化。该例子所使用的配置如下:

最大检查尝试次数：3
 正常检查间隔时间：每 3 分钟
 重试检查间隔时间：每 1 分钟

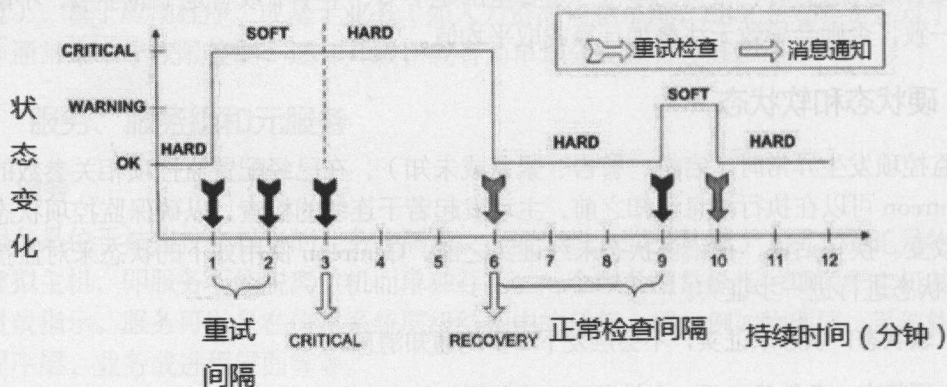


图 9-18 Centreon 的检测重试与通知机制

9.5.4 状态波动与状态特殊震荡

基于 Nagios 的调度程序有一个功能，即可以检测主机或者服务监控项的状态振荡。

振荡是指主机或者服务监控项的状态变化过于频繁。当 Nagios 探测到监控项的状态变化时，会计算状态变化的百分比。当该百分比超过了定义的高位阈值（振荡百分比高位阈值）时，Nagios 会将主机或服务监控项标记为震荡状态（即 FLAPPING 状态）。当百分比一直在低位振荡阈值持续，未超过高位阈值时，这些主机或服务状态就会被标记为正常。

Nagios 检测状态震荡的主要目的是限制虚假状态导致的频繁发送告警通知消息。事实上，由于以上特殊状态的存在，检测项通知消息的接收方仍旧有可能会收到一两个通知消息，但只要检测项未出现 Warning 或者 Critical 告警信息，此类通知消息是可以被忽略的。

在 Centreon 的 Web 用户界面中，状态变化的百分比可以在主机和服务的细节界面中查看。振荡检测的上限和下限既可以在 Nagios 调度进程的系统全局范围参数中设定，也可以在每个主机和服务监控项的个体配置中进一步指定。

提示：需要指出的是，Nagios 检测状态震荡的算法相当复杂，主要是基于最近 21 个状态的变化，再以过去一段时间内的状态报告作为权重而算出的，详细的关于状态震荡算法的情况可以参考一下 Nagios 的相关资料。

第 10 章

Centreon 的实时监控

实时的概念在 Centreon 中的体现主要基于如下两点：

- (1) 使用 Nagios 所实时采集数据的新鲜程度。
- (2) Centreon 展示这些监控数据所需要耗费的时间。

10.1 专注于实时监控的 Centreon

图 10-1 总结了 Centreon 显示实时监控数据的流程，考虑到了延迟等相关因素。

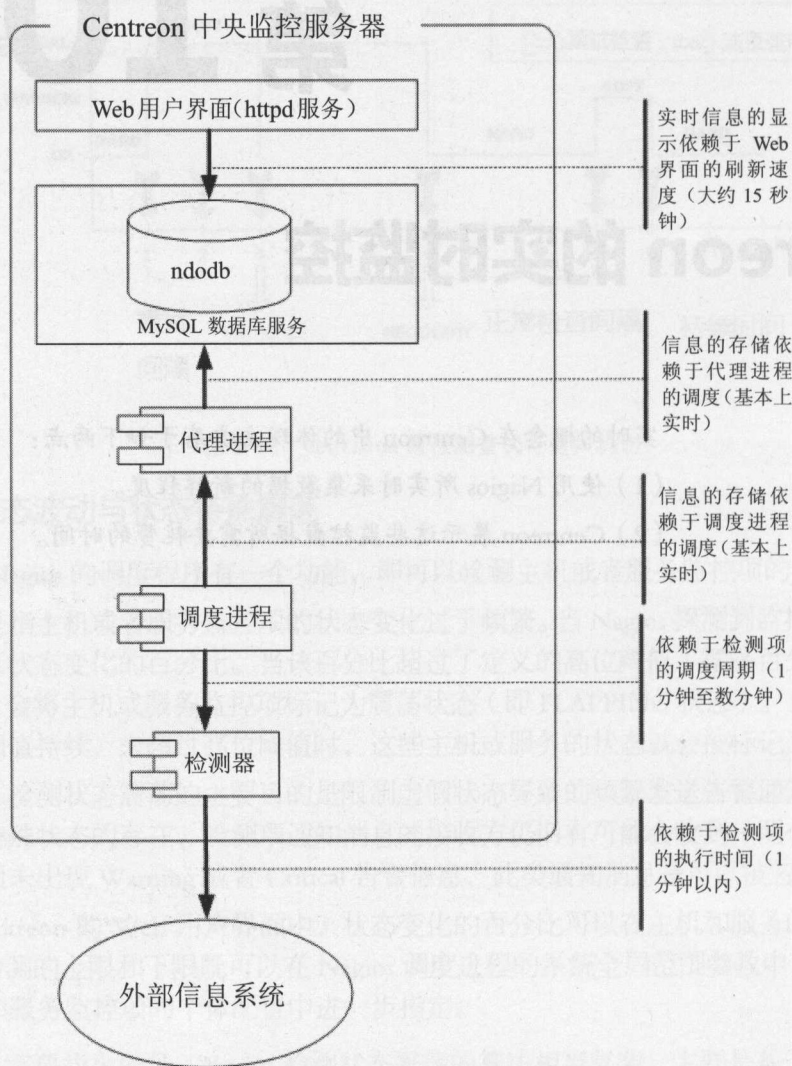


图 10-1 Centreon 显示实时告警信息的流程

由图 10-1 可以看出，影响 Centreon 系统实时显示监控信息的最重要的因素就是与检测项有关的调度延迟和检测执行延迟，两者加起来通常要占用 1 到数分钟。因此，有关检测周期的设定和有关检测延时的评估就显得相当重要（参考图 12-19），良好的检测粒度设定会在允许监控数据实时更新的同时，无需占用太多服务器资源。

在临时减少检测延迟方面，Centreon 存在以下两种机制。第一种机制是允许用户主动执行强制的，对于一个或多个主机或服务的强制检测，这种操作会覆盖系统原有的默认调度，这个操作在本章的后续部分会讲到。第二种机制是强制进行界面刷新，在接下来会讲到。

1. 在用户界面中设置刷新闻隔

一般来说,刷新 Centreon 的 Web 用户页面在物理上会对应一个应用程序重新访问数据库的操作和一个界面重绘操作。

如图 10-2 所示, Centreon 可以对界面的刷新闻隔做设定,进入菜单 Configuration→Options→Options (左侧菜单),找到 Refresh Properties 相关项。

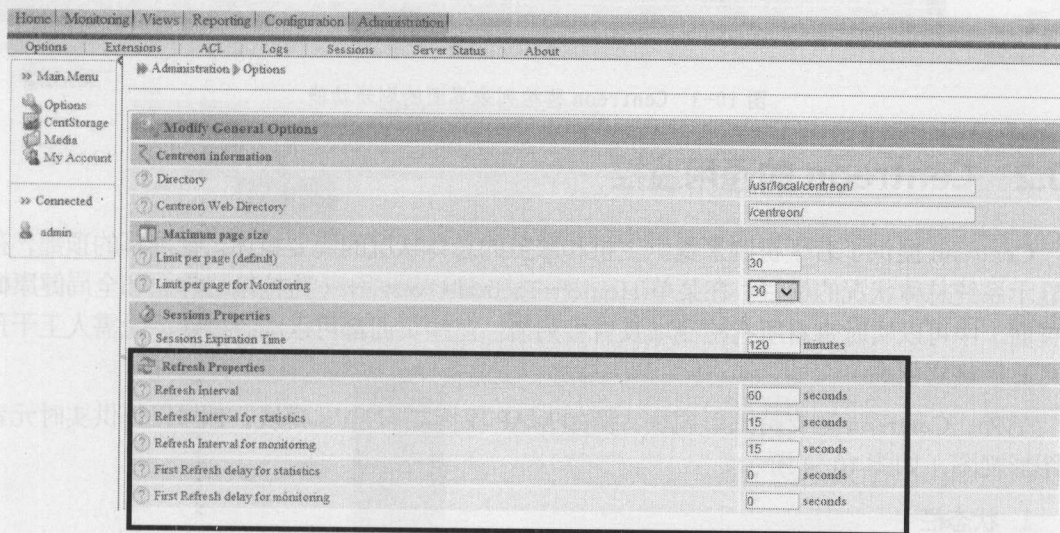


图 10-2 Centreon 界面刷新参数设定

- **Refresh Interval:** 该配置项是默认界面的整体刷新时间间隔,默认为一分钟。
- **Refresh Interval for statistics 和 Refresh Interval for monitoring:** 这两项对应的是实时监控画面的刷新间隔,一般延迟较短,默认 15 秒,分别是状态统计信息刷新间隔以及主机和服务页面监控信息的刷新间隔。
- **First Refresh delay for statistics 和 First Refresh delay for monitoring:** 这两项是用户登录后, Centreon 用户界面首次的刷新间隔,缺省值为 0 秒,即用户登录 Centreon 系统后立即执行 Centreon 用户界面的刷新操作。

注意:刷新时间间隔过短,会对监控平台的性能消耗有显著影响,特别是在连接的用户数较多的情况下。

2. 动态列表界面的刷新

如图 10-3 所示,在 Centreon 的实时监控列表界面中,存在如下 3 个按钮,用来控制列表信息的动态更新和定期更新。

- 第一个按钮 (刷新) 的作用是强制进行列表更新,从而在数据库中执行一个查询来刷新数据,并重绘列表。
- 最后一个 (暂停) 键,作用是暂时冻结列表数据的更新,使其状态保持在当前状态,便于阅读详细特定的错误消息。
- 中间的按钮 (播放) 作用是在列表更新冻结后,重新激活列表数据的自动更新。

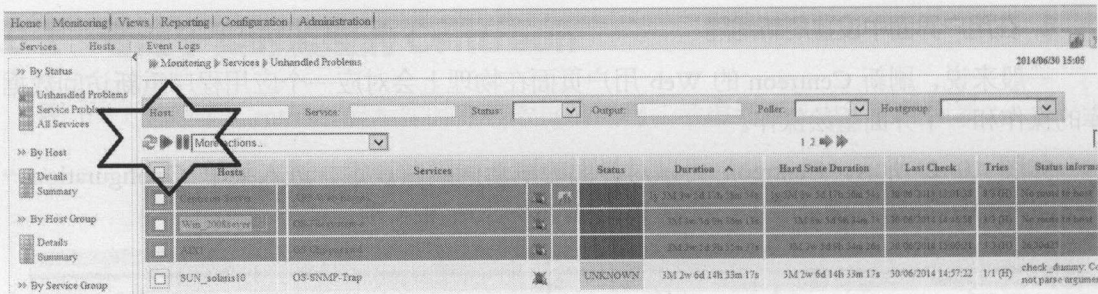


图 10-3 Centreon 监控列表界面的刷新功能

10.2 Centreon 的通用监控

Centreon 提供了若干机制来提供主机和服务的总体状况的简要概述。在屏幕的顶部，存在显示系统总体状况的横栏。在菜单 Home→Tactical Overview（监控策略界面和全局健康概览页面）中可以实时查看所有检测项及告警列表。这些页面都可以动态刷新，无需人工干预即可显示最新状态。

另外，Centreon 还支持图形模块，例如 MAP 模块或 NagVis 模块，都可以提供实时元素的图形视图，如图 10-4 所示。

1. 状态栏

Poller States	Hosts	Up	Down	Unreachable	Pending	Services	Ok	Warning	Critical	Unknown	Pending
	7	5	2	0	0	88	40	3/4	29/39	3/5	0

图 10-4 状态栏

状态栏由三部分组成：

- 状态搜集器；
- 监控主机的状态；
- 监控服务的状态。

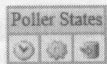


图 10-5 状态搜集器图标

2. 状态搜集器

如图 10-5 所示，状态搜集器发生错误时，三个图标会变成红色。左边的图标显示搜集器的延迟；基于代理进程搜集的数据信息，中间的图标显示调度进程是否正常运行；最后一个图标指示数据库代理进程是否正常更新后台数据库。

3. 主机和服务的监控状态

该部分可以让用户查看那些处于“已经确认告警的”和处于“正等待确认告警（Waiting 状态）”状态的主机和服务。也就是说，处于等待确认状态的主机和服务也被计入在内，但未被证实，需要等待 Nagios 执行重试机制以便确认最终状态（参考 9.5.3 小节）。单击其中一个图标，会跳转到显示相应状态的主机或服务列表。

处于错误状态的主机或者服务可以通过一个“/”符号隔开的两个数字表示。例如，2/3，第二个数字项 3 表示错误总数，第一个数字项 2 表示未确认的，以及不再与告警主机或者服务相关联的错误的数量（例如已经被用户通过 Acknowledge 动作确认过的主机或者服务）。

这种显示机制允许用户快速查看正在产生的新通知和告警的数量。

10.3 状态总揽视图

状态总揽视图，如图 10-6 所示，即菜单项 Home→Tactical Overview 是登录到 Centreon 的 Web 界面后显示的第一个画面。它提供了告警主机和服务的简要概述和状态总揽。

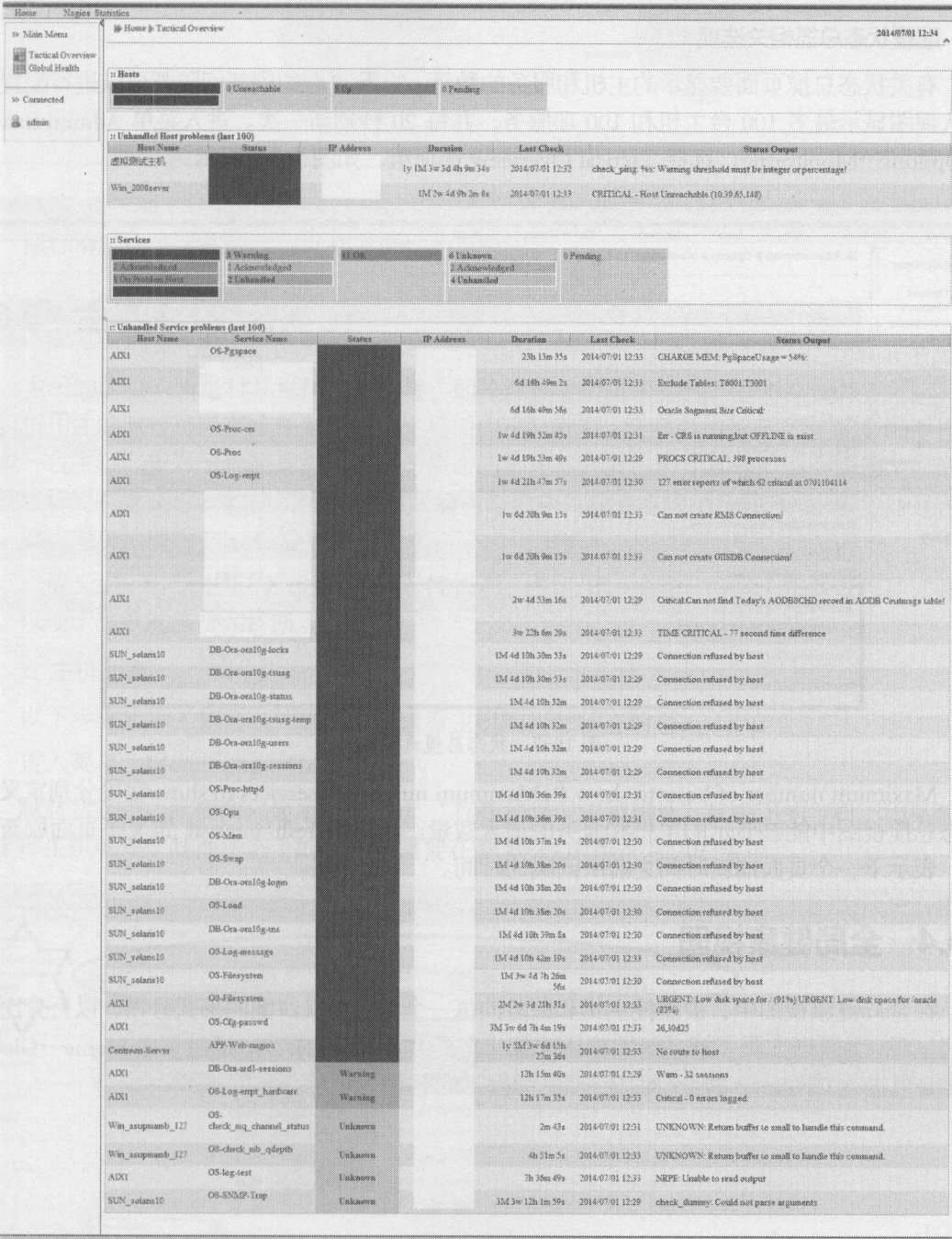


图 10-6 状态总揽视图

在图 10-6 的状态总览视图中，Centreon 将未处理的（未经用户确认 Acknowledge 和未恢复为 OK 状态的）告警主机和未处理的告警服务以分类列表的形式显示出来。每一项列表均显示了告警的详细信息，包括主机名、服务项、IP 地址、报错信息等，且点击主机名或服务项可进入相应的细节界面中，供进一步查看。

状态总览视图支持动态更新的，用户不需要手工刷新即可查看系统的最新状态，可用于业务监控视图。

配置状态总揽相关选项

有关状态总揽页面要显示的主机和服务的数量，以及页面的刷新闻隔都可以进行配置。默认视图显示最多 100 台主机和 100 项服务，并每 20 秒刷新一次。进入菜单 Administration→Options→Monitoring，选择 Tactical Overview 选项组，如图 10-7 所示。

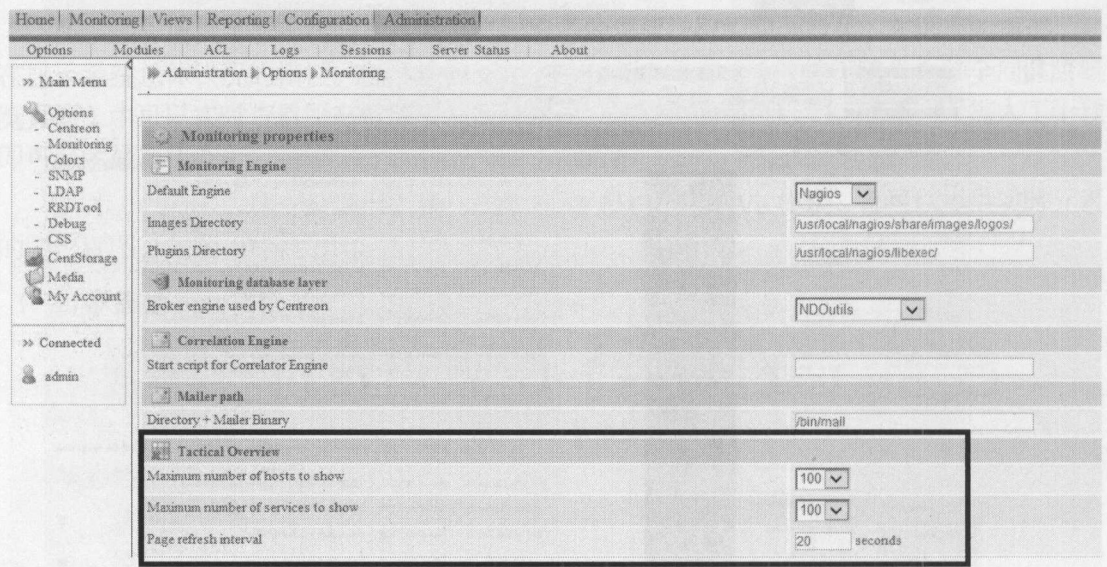


图 10-7 状态总揽选项组

Maximum number of hosts to show 和 Maximum number of services to show 字段分别定义在状态总览视图中能够显示的主机和服务的最大数量。Page refresh interval 定义了页面刷新闻隔，表示下一个页面刷新时间之前所间隔的时间。

10.4 全局健康视图

在全球健康视图中，被监控主机和服务的每一个状态都以饼图的形式分布，以百分比的形式呈现，为管理人员了解系统的整体健康状态提供了直观的视图。通过菜单 Home→Global Health 可进入 Centreon 的全局健康视图界面，如图 10-8 所示。

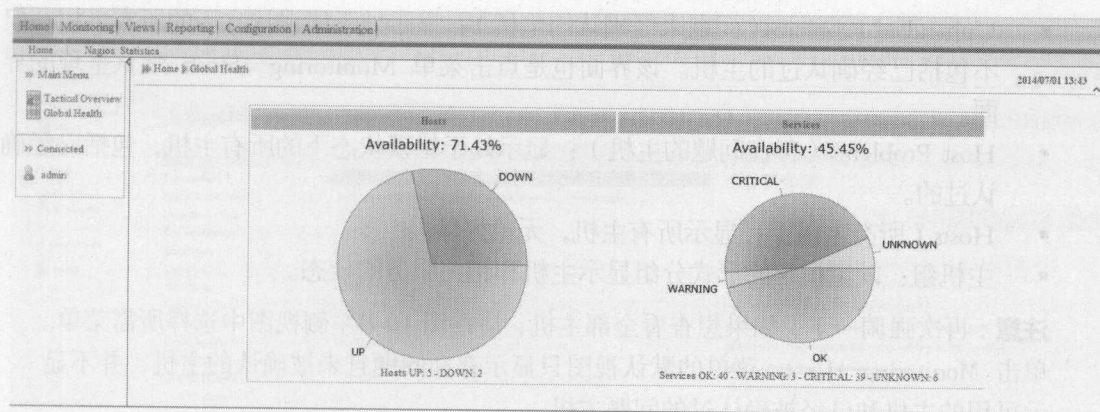


图 10-8 全局健康视图

鼠标滑过的每一部分，可以知道确切的百分比。

10.5 主机的实时监控

主机的监控一般是对其可用性的监督，即检测该主机是否存活、是否在线并正常运行。主机可用性的监控可以直接在主机的定义界面中进行配置。大多数情况下，会通过 Ping 命令检查一台主机。但对于承担某些特殊服务使命的服务器而言，使用其他的命令检测主机的可用性也许更有意义，例如使用检查 HTTP 服务是否正常的命令检测远程 Web 服务器是否可用，就比简单地使用 Ping 命令要灵活得多。

一般来说，对于主机的实时监控可以按状态、主机组、关键字检索主机并查看其详细信息，以便于后续的维护和诊断。

1. 主机列表

以下步骤显示了如何以列表的形式查看主机的实时状态信息：

进入菜单（Monitoring→Host）。

默认状态下，此视图只显示那些处于“存在错误但未被确认”状态下的主机。如果想显示全部主机或处于其他状态下的主机，需要在图 10-9 中单击左边的菜单。

Hosts	Status	IP Address	Last Check	Duration	Tries	Status Information
ADIX1	UP	10.10.10.1	02-07-2014 12:25:37	1s 25m 42.24s 12s	1/3 (H)	OK - Ping OK - Packet loss = 0%, RTA = 0.27 ms
Centreon-Server	UP	127.0.0.1	02-07-2014 12:26:01	1y 1M 2w 1d 14h 15m 17s	1/3 (H)	PING OK - Packet loss = 0%, RTA = 0.04 ms
SUN_snlms10	UP	10.10.10.10	02-07-2014 12:25:02	1M 5d 10h 30m 53s	1/3 (H)	PING OK - Packet loss = 0%, RTA = 0.47 ms
Win-test1	UP	10.10.10.10	02-07-2014 12:26:36	2d 2h 34m 50s	1/3 (H)	PING OK - Packet loss = 0%, RTA = 0.51 ms
Win_asupmmb_127	UP	10.10.10.127	02-07-2014 12:26:36	2d 2h 34m 13s	1/3 (H)	PING OK - Packet loss = 0%, RTA = 0.42 ms

图 10-9 主机监控列表

- Unhandled Problems (问题未经确认的主机)：显示处于错误状态下的所有主机，但不包括已经确认过的主机。该界面也是点击菜单 Monitoring→Hosts 默认呈现的界面。
- Host Problems (存在问题的主机)：显示处于错误状态下的所有主机，包括已经确认过的。
- Hosts (所有主机)：显示所有主机，无论好坏。
- 主机组：以主机组的形式分组显示主机和相关服务的状态。

注意：再次强调一下，如果想查看全部主机，请在图 10-9 左侧视图中选择所需菜单。单击 Monitoring→Hosts 菜单的默认视图只显示存在问题且未被确认的主机，并不显示可用的主机和已经被确认过的问题主机。

2. 过滤器

在主机列表中，用户可以根据若干标准对主机进行筛选，以便于进一步定位主机。Host 字段支持按照主机名称关键字和 IP 地址定位主机，必须输入至少 3 个字符，后台的自动搜索动作才能被激活。其余过滤器包括：按照主机的状态搜索。按照轮询器搜索（用于分布式架构）和按照主机组搜索，如图 10-10 所示。

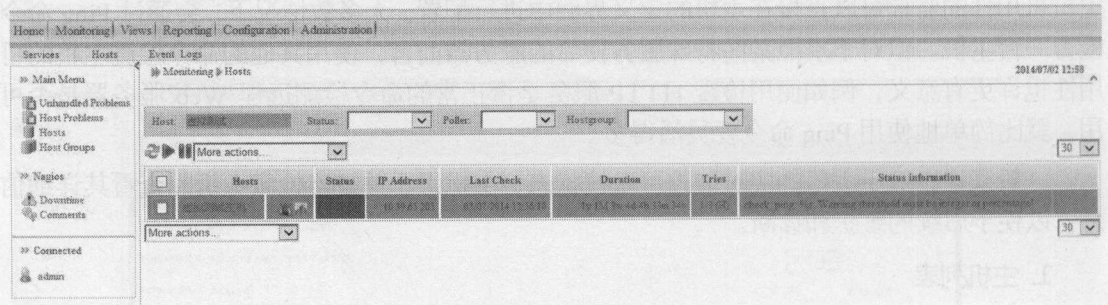




图 10-10 主机过滤

主机列表中包含有详细的下一段信息：

-  此图标表示此主机禁用通知，即出现问题之后不会对外界发送短信、邮件等通知消息。
-  此图标表示该主机具备性能曲线图，单击该图标可以直接访问该主机的性能图形。

10.6 主机的详细信息视图

在实时监控列表中点击某一个主机的名称，可以跳转到该主机的详细信息界面，如图 10-11 所示。

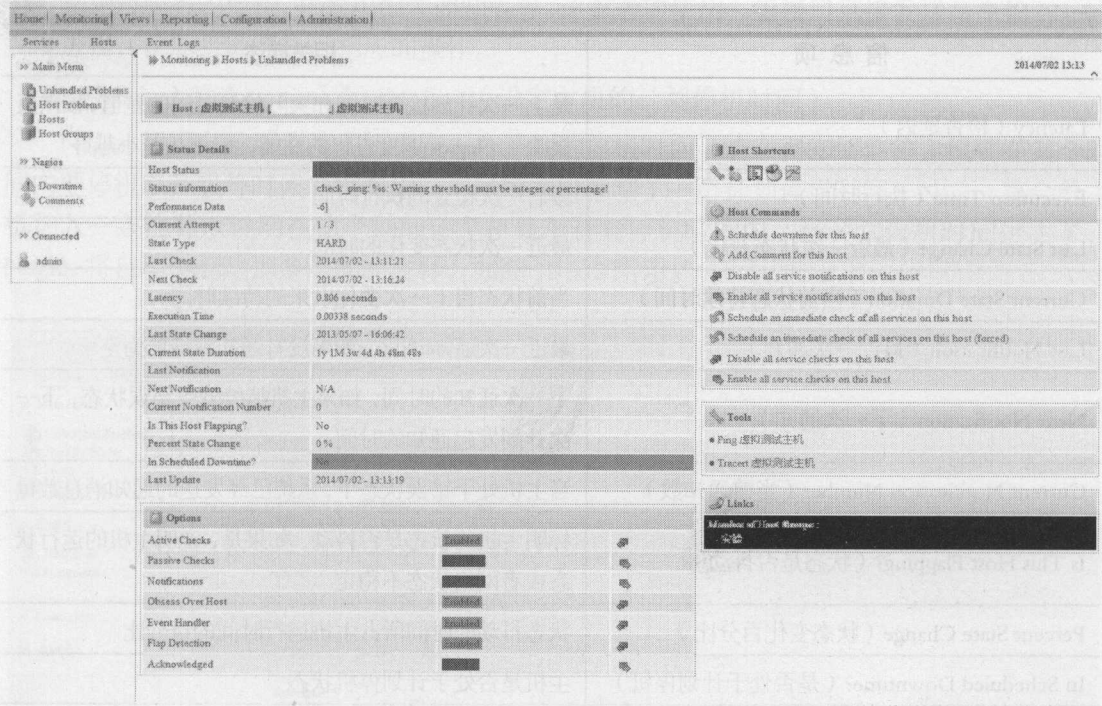


图 10-11 主机的详细信息

1. Status Details

图 10-11 显示的主机信息非常全面。在屏幕的顶部依次显示出主机名、IP 地址和该主机的别名，格式为<host NAME>[<IP>|<Alias>]。该界面中的相关信息在表 10-1 中有说明。

表 10-1 主机详细信息列表项

信 息 项	相关描述
Host Status（主机状态）	主机是否可用 – UP：可用 – DOWN：不可用
Status information（状态详细信息）	显示的是远程检测插件返回的自由文本信息，该信息的格式可以在编写检测插件时指定输出格式
Performance Data（性能数据）	用于构建性能图形，在后续的有关通知、图形和报表的章节中会讲到
Current Attempt（当前重试次数）	当主机处于“软”状态下的重试次数，格式如下：当前重试次数/重试总数
State Type（状态类型）	软状态或者硬状态
Last Check（上次检查）	上一次对该主机检查的时间
Next Check（下次检查）	下一次计划的对该主机检查的时间






续表

信 息 项	相关描述
Latency（检查延迟）	是上一次计划检测时间和实际检测时间的差值,该差值反映了 Nagios 调度进程的性能,应该是越小越好
Execution Time（执行时间）	最后一次检查的执行时间
Last State Change（最近一次状态变化）	最近一次状态变化的时间
Current State Duration（当前状态持续时间）	当前状态自上一次变化以来的持续时间
Last Notification（最近一次通知）	最近一次通知时间,如果没有通知则该项为空
Next Notification（下一次通知）	下一次通知的时间,如果主机持续处于错误状态,下一次计划发起通知的时间
Current Notification Number（当前通知数）	当主机处于错误状态下,系统已经发送的通知消息数量
Is This Host Flapping?（状态是否抖动）	标明主机的状态是否抖动,如果是,说明主机的运行状态或者网络状态不稳定
Percent State Change（状态变化百分比）	状态抖动的时间所占主机运行时间的百分比
In Scheduled Downtime?（是否处于计划停机）	主机是否处于计划停机状态
Last Update（最近一次更新）	当前页面所显示的主机详细信息的最近一次更新时间

2. Host Shortcuts（到其他界面的快捷方式）

在主机详细信息页面中有指向其他页面的快捷按钮链接,如表 10-2 所示。

表 10-2 主机详细页面的快捷按钮

图 标	简 述	菜 单 项
	主机配置页面	Configuration→Hosts
	查看该主机的所有监控项	Monitoring→Services→All Services
	查看该主机的告警日志	Monitoring→Event Logs→ All Logs
	查看该主机的监控报表	Reporting→Dashboard
	查看该主机的性能图形	Views→Graphs

3. Options 管理选项

该组选项允许您在线更改被监控主机的一些检测选项,并即时生效,例如,禁用 Nagios 调度进程对于该主机的主动、被动检查,禁用该主机的通知等等。在该组选项中所做的操作不会写入后台数据库中,这就意味着重启服务器后,之前所做的配置修改会丢失。

4. Host Commands 选项（主机命令选项）

在一个复杂的 Centreon 和 Nagios 监控系统中,需要提供一些灵活的接口来对主机或者服务控制。如:当某台主机宕机,需要禁用对改主机的检测;再如某个服务暂时进行维护,

需要禁用对服务的检测,并禁用相关通知等等。该组选项使用一组接口直接发送命令给 Nagios 调度进程,使后者能够执行相应的操作。

(1) Schedule downtime for this host (设定主机的计划停机时间)

Nagios 里可以给所监控主机指定一个计划的停机时间。这在得知所监控主机要在某个时间内要停机以升级等时候非常有用。一旦给主机编制了一个计划停机时间, Nagios 将会给主机加入一条注释以说明在这个期间该主机处于计划停机时间内。当计划停机时间过去了, Nagios 将自动地删除那条添加的注释,如图 10-12 所示。

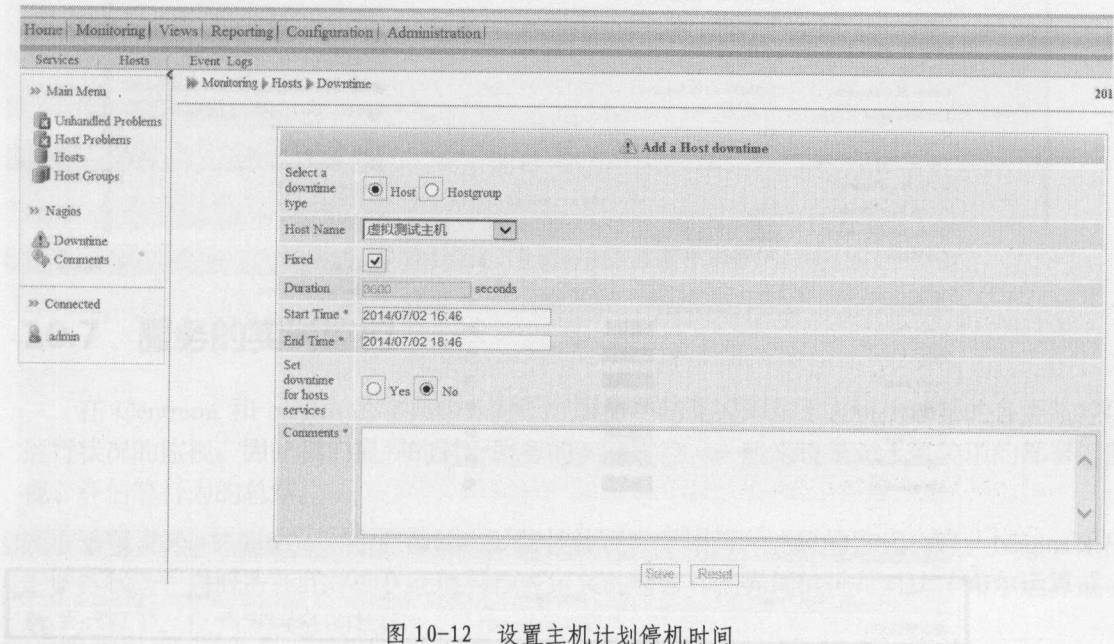


图 10-12 设置主机计划停机时间

当通过 Web 界面来编制一个主机与服务的计划停机时间时, Centreon 会询问停机时间是固定式还是可变式(即图 10-12 中的“Fixed”)。

- “固定式”即停机时间启动和停止时刻会严格按照用户所编制计划时所设定的时间内开始与结束。
- “可变式”停机时间可以用在当知道主机与服务要停机 X 分钟(或 X 小时)但是并不知道什么时候开始停机时。当使用可变式停机时间, Nagios 将在某个时间开始执行停机,到用户指定的时间间隔达到后结束停机。它假定了主机与服务使用一个可变的停机时间段来做停机时的操作,而这个停机时间段开始于系统真正探测到的状态——主机进入实际而非计划的宕机(或不可达)状态或是服务处于非正常状态时。而结束时间则不是基于系统的判断,而是严格基于指定的停机时间间隔之后的那个时间点,即便是在此之前主机与服务已经恢复,系统也是认为它还处于停机时间内。

(2) Add Comment for this host (为主机添加注释)

为该主机添加一条注释,该注释信息会显示在主机详细信息界面中,如图 10-13 所示。

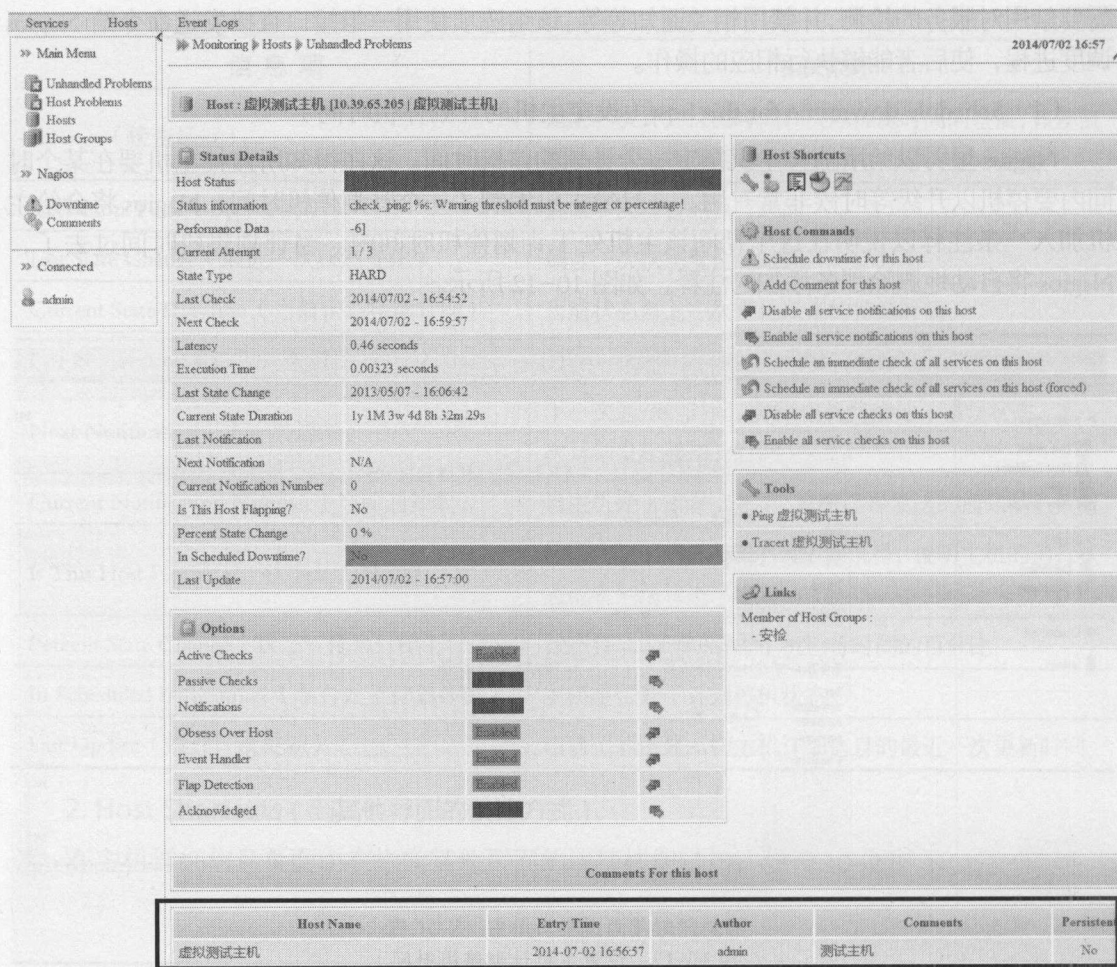


图 10-13 为主机添加注释

- Disable all service notifications on this host
关闭该主机上所有服务的告警通知。
 - Enable all service notifications on this host
启用该主机上所有服务的告警通知。
 - Schedule an immediate check of all services on this host
发起对于该主机上的所有监控项的检测。
 - Schedule an immediate check of all services on this host (forced)
强制立即进行该主机上所有监控项的检测。
 - Disable all service checks on this host
禁用该主机上所有监控项的检测。
 - Enable all service checks on this host
启用该主机上所有监控项的检测。
- 以上命令大多是对该主机上部署的相关服务监控项的调度、启用、禁用命令，不具备相

关图形界面，仅是通过 Web 界面向 Nagios 调度进程发送命令，由后者负责执行。

5. Tools (检测主机的工具)

该组选项提供了对于监控主机是否可用的两种检测工具，分别是 Ping 命令和 Tracert 工具，用以检测主机是否在线、路由是否正确。单击命令按钮可以在 Centreon 中央服务器上执行相关检测命令，在弹出窗口中会有命令执行的结果，如图 10-14 所示。

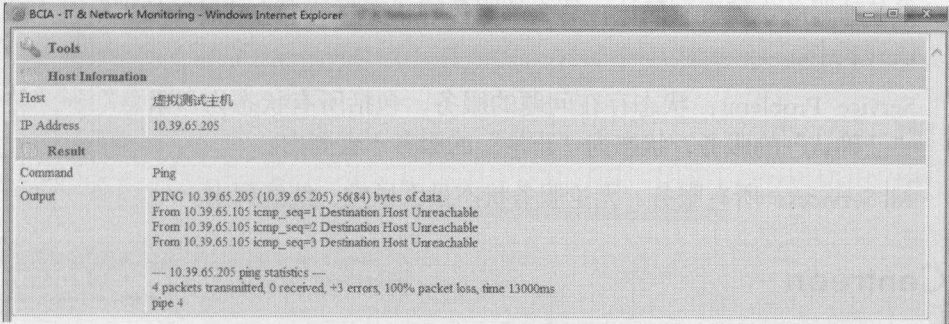


图 10-14 主机检测命令执行界面

10.7 服务的实时监控

在 Centreon 和 Nagios 的字典中，对于“服务”的监控即对于主机上部署的各类监控项运行状况的监视。因此我们提到的对于服务的实时监控，一般来说是对于监控项的部署、监视、评估等行为的总结。

一般说来，Centreon 对于服务的实时监控体现在采用瞬时测量的方式，基于 Nagios 采集的服务状态，将服务在此时的状态值与预先定义的警告和临界阈值进行对比（布尔运算或者数学运算），以判定服务的状态。

Centreon 可以以列表的形式显示服务的瞬时状态，支持按照关键字、主机、服务状态等字段过滤和查找相应服务，并查看服务的详细信息，以上步骤都可以在 Web 界面中执行。

1. 查看监控服务列表

在 Centreon 中查看服务列表，可采取如下方式：

在 Centreon 的 Web 用户界面中，单击菜单 Monitoring→Services，进入图 10-15 所示界面。

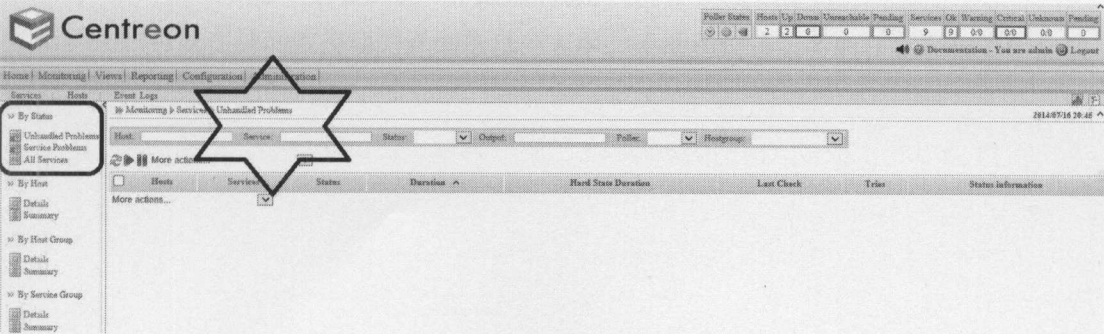


图 10-15 查看默认进入的“未解决问题”服务列表

如上图 10-15 所示，系统默认进入的监控服务列表界面是 Unhandled Problems（问题未处理）服务列表，如果 Centreon 中部署的所有监控项状态均为正常，那么该列表界面将会是空白。

Centreon 的左侧菜单提供了基于服务状态的服务列表过滤查看功能，如图 10-16 所示。

- **Unhandled Problems:** 在服务列表中列出处于“问题未处理”状态的服务。注意如果某项服务告警得到确认，那么该服务的状态将被视为“已经处理”，不会出现在该列表中。
- **Service Problems:** 状态存在问题的服务，包括所有状态为“紧急”、“警告”和“已确认”的服务，状态为“正常”的服务不会列出。
- **All Services:** 所有服务，无论服务状态是否异常，统统列出。

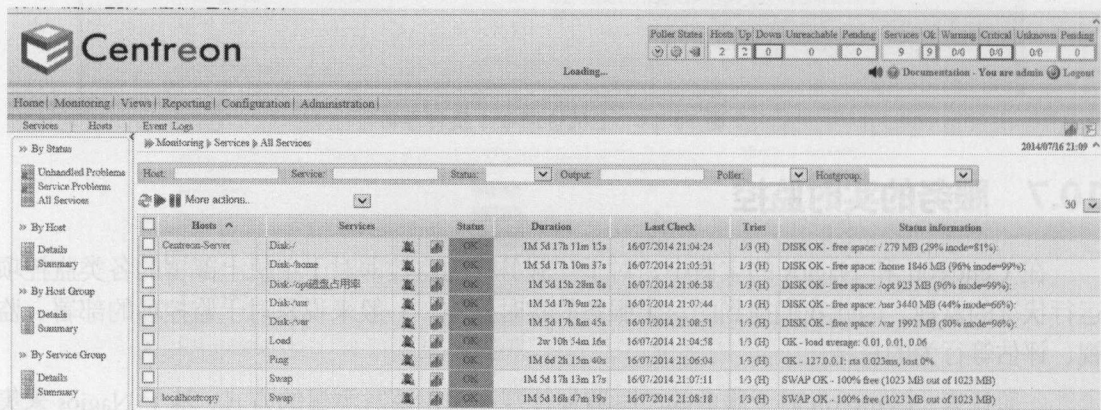


图 10-16 所有服务列表

您可能已经注意到，在每一个服务的 Status 前均存在两个图标，分别具备如下功能：
图标的含义是显示该服务项未启用通知，如果对于服务项启用通知，那么该图标将会消失。
图标的含义是该服务存在性能图形，单击图标可以进一步查看该服务的性能趋势图。

2. 查看服务的详细信息

单击服务列表中任一服务的名称链接，可跳转至该服务的详细信息界面，如图 10-17 所示。

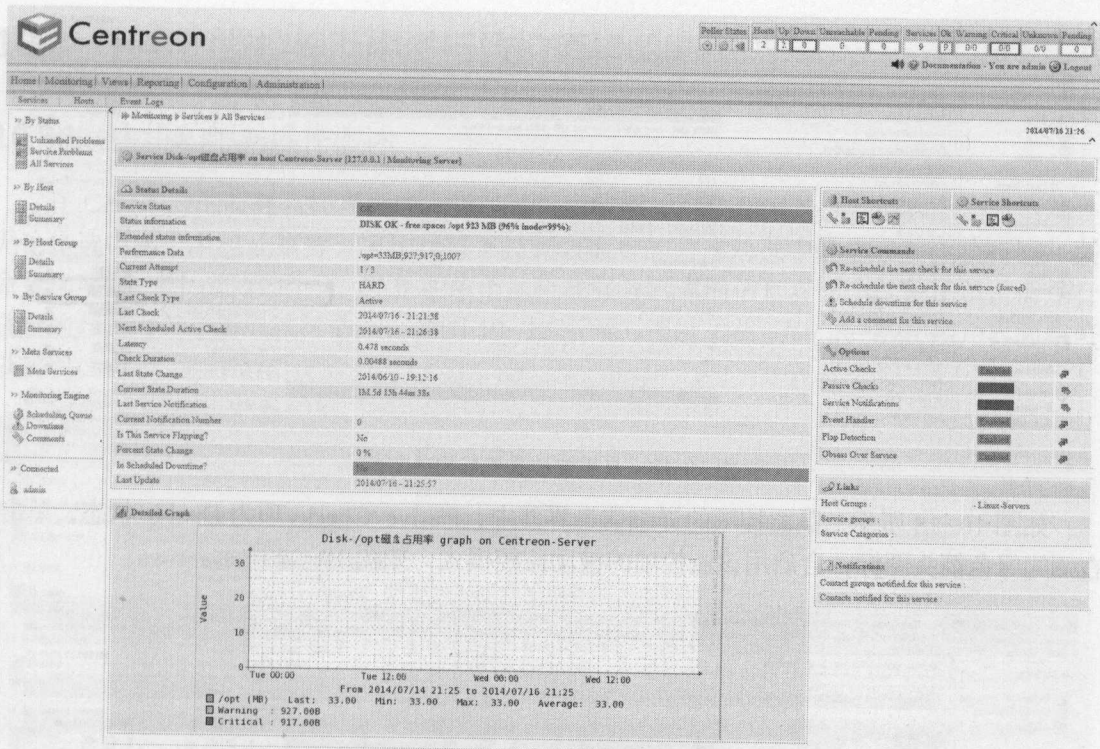


图 10-17 服务的详细信息界面

上图 10-17 显示的服务详细信息界面和图 10-11 显示的主机详细信息界面基本类似，且快捷方式图标也基本一致。

如果该服务的检测项能够返回性能数据，那么在服务详细信息界面中会出现 Detailed Graph 项，即能够显示该服务的性能图形，显示时间段为 2 天；如果检测探针未能返回性能数据，那么该性能图形栏将不会显示。

3. 有关被动服务的相关说明

与主动服务一样，被动监控项服务（Passive Service）同样会出现在如上图 10-16 所示的服务列表中。但是被动监控服务具备如下特征：

- “软状态”的概念不适于被动监控项，一旦被动监控项告警，将被判断为处于“硬”告警状态，将会触发一个通知消息。
- 用户可以在被动监控项的详细信息列表中，手动指定该服务的状态和相关附加信息，如图 10-18 所示。

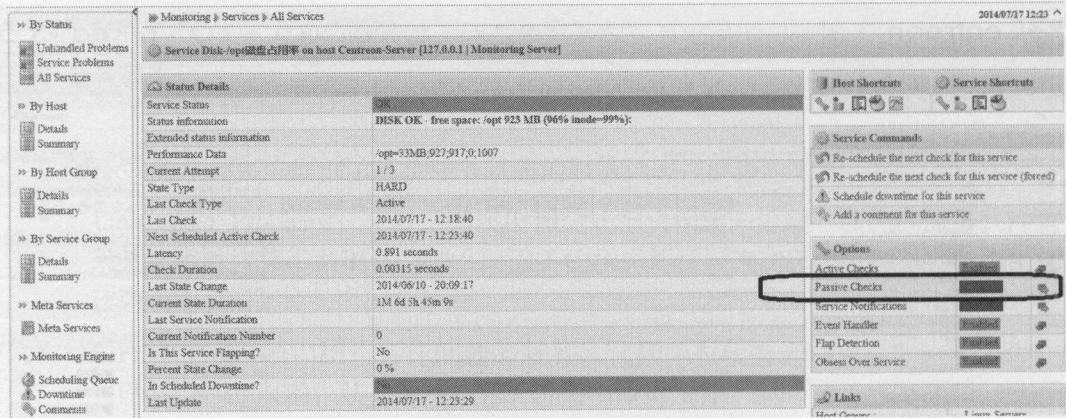


图 10-18 启用被动服务监控项之前

如图 10-18 所示，某监控项的详细信息界面中，Passive Checks 项为 Disable 状态，单击右侧的绿色箭头图标，会启用该监控项的被动监控模式，如图 10-19 所示。

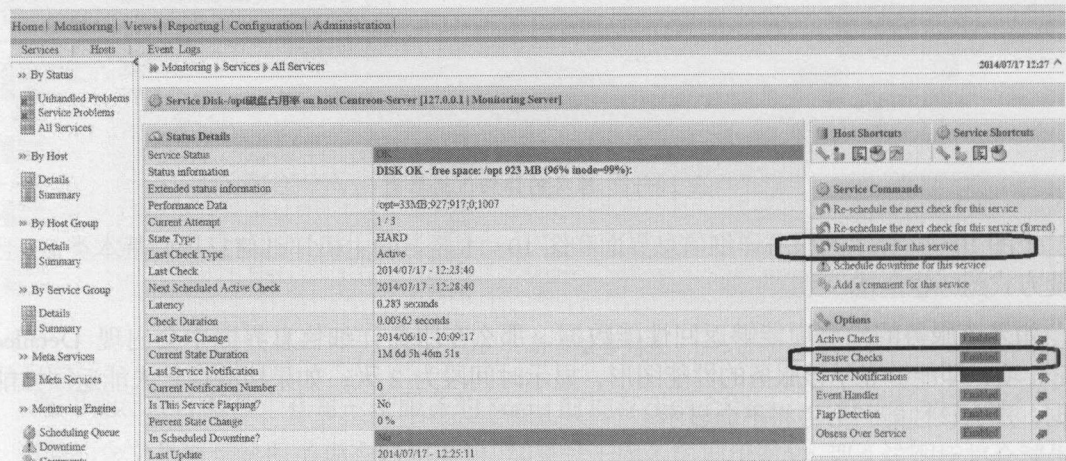


图 10-19 启用被动监控项之后

如上图 10-19 所示，启用该监控项的被动检测模式之后，在 Service Commands 列表中，会出现 Submit result for this service 命令，意味着可以为该服务提交被动检查结果，如图 10-20 所示。

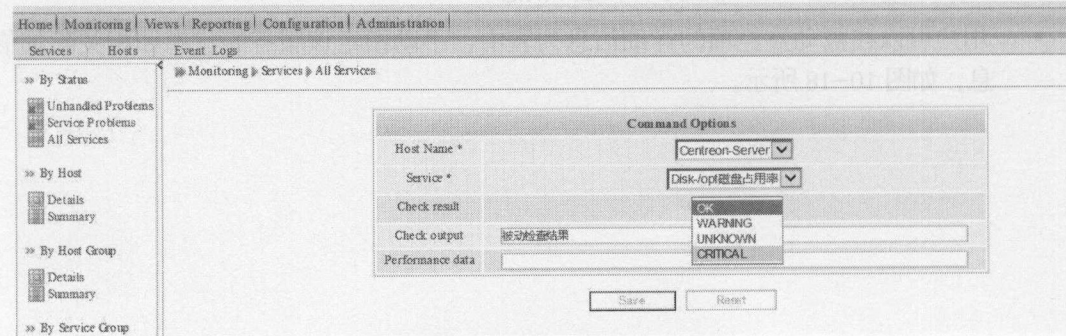


图 10-20 提交被动检查结果

10.8 在实时监控界面中进行监控项相关操作

10.8.1 主机和服务操作概述

在 Centreon 的实时监控界面中，提供了多项能够对主机和服务进行操作的命令选项。无论是选中多个主机或服务，采用执行下拉列表命令的方式对选中的多个对象执行同一条命令选项，如图 10-21 所示，还是在单个主机或服务的详细信息界面中，采用单击命令链接的方式对单个对象执行命令，如图 10-22 所示，通过 Centreon 的 Web 用户界面都可以实现。

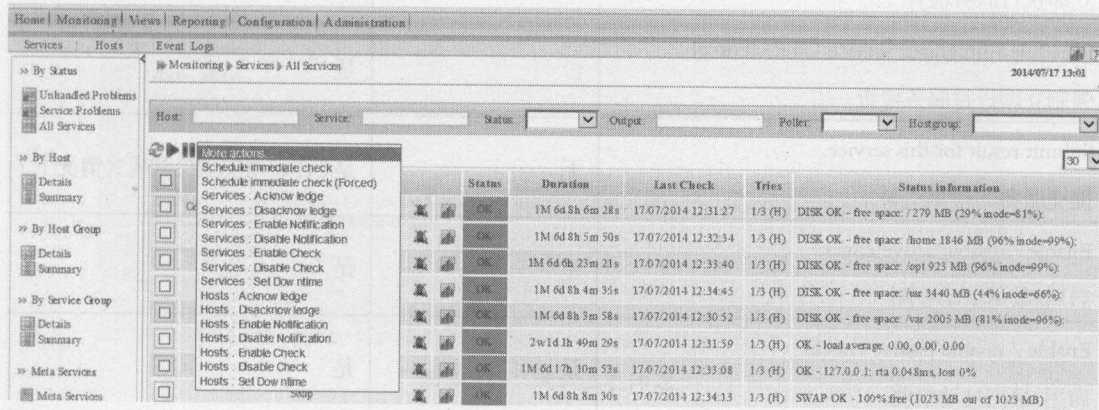


图 10-21 实时监控界面中的下拉式命令列表

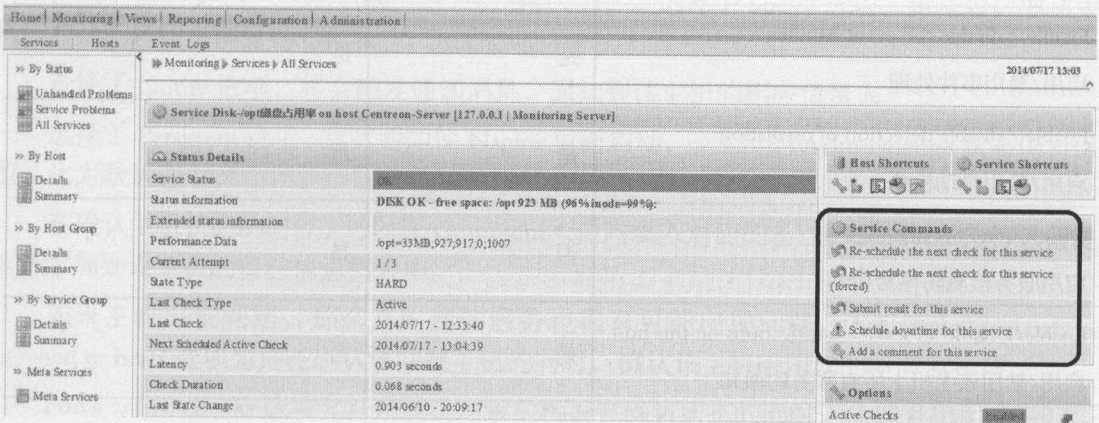


图 10-22 详细信息界面中的清单式命令列表

表 10-3 列出了主机和服务监控项所支持的命令的相关解释，注意某些链接只有当相关选项生效后才出现。

表 10-3 主机和服务监控项命令行列表






命令选项	主机是否支持	服务是否支持
Acknowledge / Disacknowledge 确认/取消确认	是	是
Add a comment 添加备注信息	是	是
Schedule Immediate Service Check 立即执行服务检查	否	是
Schedule Immediate Service Check(Force) 强制立即进行服务检查	否	是
Submit result for this service 提交服务的检测结果	无	是 (在启用被动检测模式情况下)
Enable / disable active checks 启用/禁用主动检测	是	是
Enable / disable passive checks 启用/禁用被动检测	是	是
Enable / disable notifications 启用/禁用通知	是	是
Enable / disable the event handler 启用/禁用事件处理	是	是
Enable / disable detection of oscillations 启用/禁用抖动检测	是	是
Obsess Over Service 启用服务检测防停滞机制	否	是
Enable/Disable all service notifications on this host 启用/禁用该主机上所有服务的通知	是	否
Set DownTime 设置停机时间区间	是	是

10.8.2 处于告警状态下的主机或者服务进行确认

在主机或者服务的状态处于警告或者紧急情况下，一旦用户做出问题确认，那么系统就认为用户已经观察到了主机或者服务目前正处于问题状态，并且已经在考虑解决方案。系统将不会再为问题主机或者服务发送告警信息，并且在图 10-15 所示的问题主机或者服务列表

中，将不会再显示已经被确认过的问题或者主机服务。

对处于告警或者紧急状态下的问题主机或服务进行人工确认，需要执行下列步骤：

- (1) 首先通过单击菜单 Monitoring → Hosts → Unhandled Problems，进入默认的问题主机列表界面，并选择需要处理的主机名称前的复选框。
- (2) 单击      下拉列表，选择要执行的命令 Hosts: Acknowledge，出现如图 10-23 所示的对话框。

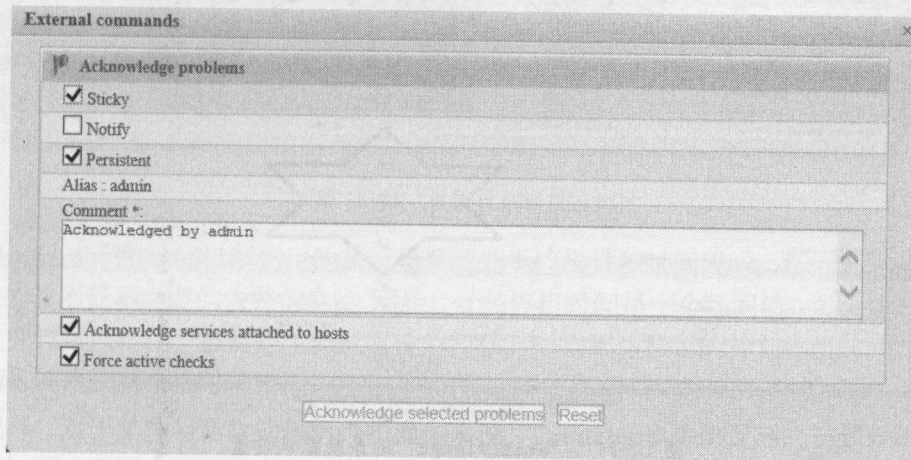


图 10-23 确认主机问题

若问题主机处于 Acknowledge 状态，如果选择了 Sticky 选项，那么当该问题主机在非 OK 状态之间，即在警告、紧急、未知、待定等状态之间迁移的时候，该主机仍将处于 Acknowledge 状态，而不会呈现相应的错误状态以待用户再次确认。选择了 Sticky 选项，意味着只要该主机没有从问题状态恢复到 OK 状态，Acknowledge 状态将一直保留。

选择了 Notify 选项，将会触发通知消息，通知到该主机的相应负责人。

选择了 Persistent 选项，意味着即使 Nagios 重新启动，该主机的 Acknowledge 状态仍将保留，此为默认选项。

在确认主机状态的时候，可以在 Comment * 栏内填入对应的备注，Centreon 会将该备注信息添加到该主机的备注信息列表中。

如果主机出现问题，那么其附属的服务也很有可能出现问题。Acknowledge services attached to hosts 选项允许在确认该问题主机的同时，确认该主机所附属的所有监控项。

Force active checks 选项允许系统在对该主机设置确认状态的同时，立即发起主动检查，以确保该主机的是处于故障状态。

有关以上选项的系统默认设置，在如图 10-24 所示的菜单项 Administration → Options → Monitoring (左侧菜单) 中可以找到：

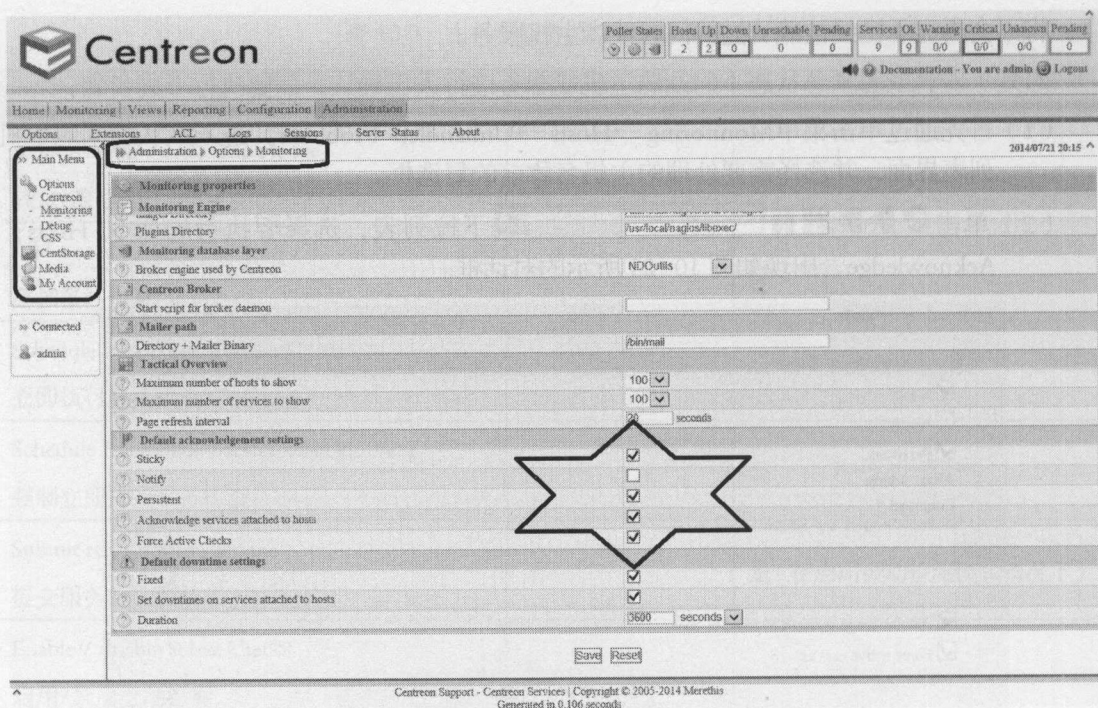


图 10-24 “acknowledge” 的系统默认设置

对于处于问题状态下的监控服务，其“确认”界面基本一致。

对已经确认的问题主机或者服务进行“Disacknowledge（取消确认）”操作，只需选中相关列表项，并选择 Disacknowledge 菜单即可。需要注意的是，取消对于问题主机的确认并不能同时取消对于其附属服务的确认操作，两者必须分开进行。

10.8.3 计划停机

1. 固定的和浮动的停机时间定义

Centreon 提供了在特定时间段内，暂停部分主机或者服务监控的机制，目的是在一个时间区间内，暂停主机和服务的检测以及相应的通知消息，这种机制称为计划停机时间。在 Centreon 的 Web 用户界面中，设置了计划停机时间的主机或者服务项的背景底色以粉红色来表示，如图 10-25 所示。

图 10-25 设置计划停机时间段

另外，对于某些经常性的停机维护操作来说，Centreon 也提供灵活的计划停机时间，使检测能够更接近于实际的停机时间点，而非之前预设的固定时间点，该机制在下面的介绍中会详细讲述。

对于 Nagios 等调度进程来说, 固定停机时间意味着需要设置一个固定的停机开始时间和一个固定的停机结束时间, 两者缺一不可。Nagios 在正式调度主机或者服务停机的時候, 也严格踩点, 不会考虑被调度主机或者服务的真实停机维护时间。

图 10-26 是采取滑动停机时间机制的示例。

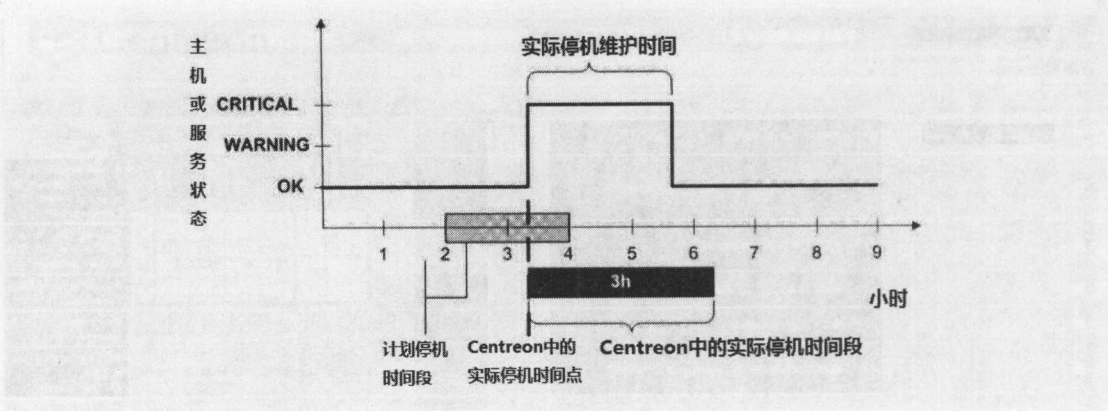


图 10-26 “滑动” 停机时间示例

2. 在 Centreon 中设置计划停机时间

在 Centreon 中设置计划停机时间可以通过以下两种途径进行：

- (1) 通过单击菜单 Monitoring，选择 Services 或者是 Hosts，在左侧的竖状菜单中，可以找到 Downtime 选项。
- (2) 在主机或者服务的详细信息页面中，也可以在 Service Commands 栏内找到 Schedule downtime for this host 或者 Schedule downtime for this host 按钮，如图 10-11 及图 10-17 所示。

3. 设置主机的计划停机时间

在 Centreon 中设置主机的计划停机时间界面如图 10-27 所示。

该图显示了 Centreon 中“Add a Host downtime”的表单。表单包含以下字段：

- Select a downtime type: ☒ Host ☐ Hostgroup
- Host Name: 下拉菜单，显示 AIX1
- Fixed: ☒
- Duration: 3600 seconds
- Start Time *: 2014/07/22 16:57
- End Time *: 2014/07/22 18:57
- Set downtime for hosts services: ☐ Yes ☒ No
- Comments *: 文本输入框

表单底部有 Save 和 Reset 按钮。

图 10-27 为主机设置计划停机时间

- (1) Select a downtime type 选项的目的是设定停机时间的对象，是对某台主机还是对主机组。

- (2) Fixed 选项的作用是设定该计划停机是采用固定停机时间机制还是“滑动”停机时间机制。默认该项为选中状态，即默认采用固定停机时间机制。
- (3) Duration 选项的作用是在采用“滑动”停机时间的机制下，设定停机时长，默认是 3600 秒。
- (4) Set downtime for hosts services 设定了在为该主机设定固定计划停机时间的同时，是否与该主机关联的所有服务都设定同样的停机时间。
- (5) Comment 栏内可以使用户为本次计划停机设置备注信息。

4. 设置服务的计划停机时间

服务的计划停机时间设置方式与主机的设置方式类似，如图 10-28 所示。

Add a Service downtime

Host Name *
Centreon-Server

Service *
OS-Load

Fixed
☒

Duration
3600 seconds

Start Time *
2014/07/22 17:30

End Time *
2014/07/22 19:30

Comments *

Save

Reset

图 10-28 为服务设置计划停机时间

5. 计划停机的查看与管理

在 Centreon 的用户界面中选择菜单 Monitoring，在左侧的竖状菜单中选择 Downtime，即可进入计划停机设置项的管理界面，在此可对之前设置的主机或者服务的停机计划进行管理，如图 10-29 所示。

Home | Monitoring | News | Reporting | Configuration | Administration

Services Hosts Event Logs

Monitoring > Services > Downtime

2014/07/22 19:36

Unhandled Problems
Service Problems
All Services

By Host
Details
Summary

By Host Group
Details
Summary

Meta Services
Meta Services

Nagios
Downtime
Downtime

Connected
admin

Add a downtime

1 2

Form 30 Page 1/2

	Host Name	Services	Start Time	End Time	Duration	Author	Comments	Started	Fixed
<input type="checkbox"/>	AIX1	OS-Cpu_1	07/22/2014 17:38	07/22/2014 19:38	7200 s	admin	Downtime set by admin	Yes	Yes
<input type="checkbox"/>	AIX1		07/22/2014 17:38	07/22/2014 19:38	7200 s	admin	Downtime set by admin	Yes	Yes
<input type="checkbox"/>	AIX1	OS-Mem	07/22/2014 17:38	07/22/2014 19:38	7200 s	admin	Downtime set by admin	Yes	Yes
<input type="checkbox"/>	AIX1	DB-Oracle-1-locks	07/22/2014 17:38	07/22/2014 19:38	7200 s	admin	Downtime set by admin	Yes	Yes
<input type="checkbox"/>	AIX1	OS-Cpu-passwd	07/22/2014 17:38	07/22/2014 19:38	7200 s	admin	Downtime set by admin	Yes	Yes
<input type="checkbox"/>	AIX1		07/22/2014 17:38	07/22/2014 19:38	7200 s	admin	Downtime set by admin	Yes	Yes
<input type="checkbox"/>	AIX1	DB-Oracle-users	07/22/2014 17:38	07/22/2014 19:38	7200 s	admin	Downtime set by admin	Yes	Yes
<input type="checkbox"/>	AIX1	OS-Log-encrypt hardware	07/22/2014 17:38	07/22/2014 19:38	7200 s	admin	Downtime set by admin	Yes	Yes
<input type="checkbox"/>	AIX1		07/22/2014 17:38	07/22/2014 19:38	7200 s	admin	Downtime set by admin	Yes	Yes

图 10-29 计划停机项的管理

在图 10-29 中的计划停机项管理界面中，可以通过 Host Name、Service 以及 Output3 类过滤项查找之前定义的计划停机项。Show finished downtime 选项指定了是否显示计划停机项的真实结束时间，该选项在查看那些采用“滑动”机制的计划停机项的真实停机时间时比较有用。

该界面同样提供了 Add a downtime 链接，用户可以用它添加计划停机项，注意为主机和服

务设置计划停机项时存在选项上的细微不同。

该界面也提供了计划停机项的删除选项，需要注意的是删除主机的计划停机项时，并不会同时删除该主机所附加服务的计划停机项，后者仍旧需要手工删除。

Centreon 中设定计划停机项时的相关默认设置在如下路径 Administration→Options，然后再单击 Web 界面左侧的竖状菜单，选择 Options→Monitoring，进入如下界面，如图 10-30 所示。

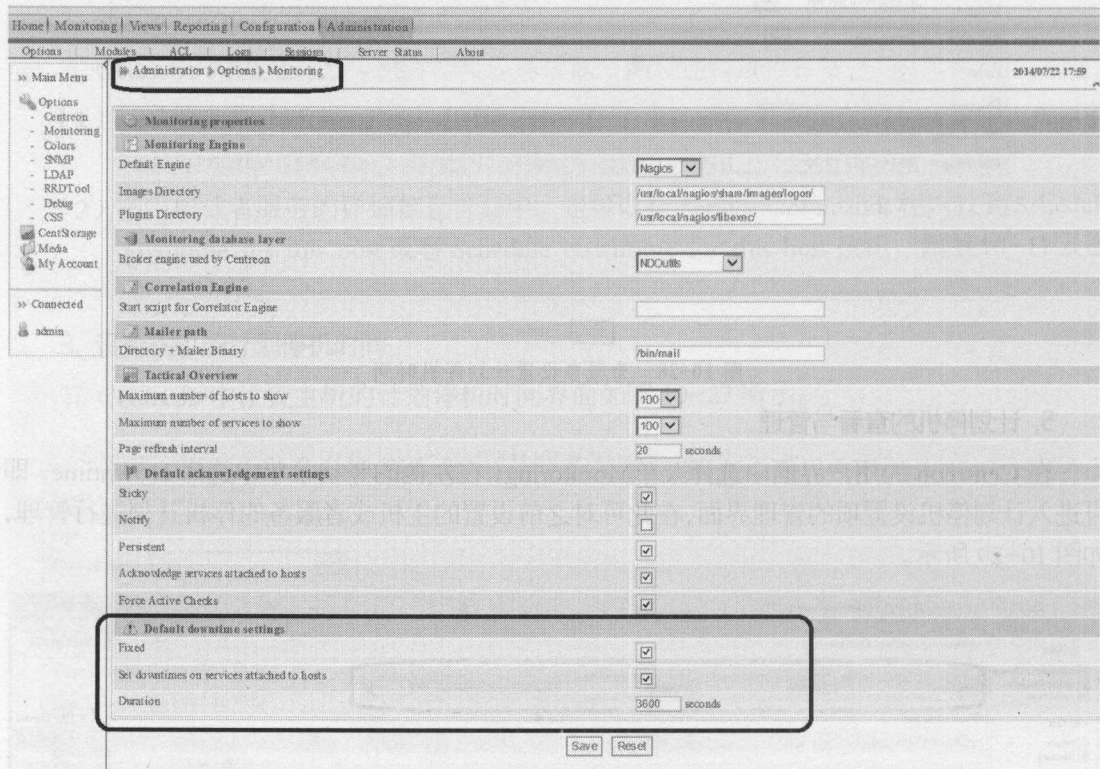


图 10-30 计划停机项时的相关默认设置

10.8.4 添加备注

在 Centreon 中，可以随时为主机或者服务添加文本形式的备注信息。除了手工添加备注外，有些备注是在对问题主机或者问题服务进行确认操作、或者对主机或服务添加计划停机时，由系统自动添加的。添加或者删除备注信息的相关操作不会立刻反映在备注项管理界面中，而是需要等待 Nagios 进程调度相关操作项，才能显示出最新结果。

添加备注信息既可以从主机和服务的实时监控列表中进行，还可以通过主机或服务的详细信息视图的操作下拉菜单中进行，也可以从专用的备注信息管理界面来完成，如图 10-31

所示。

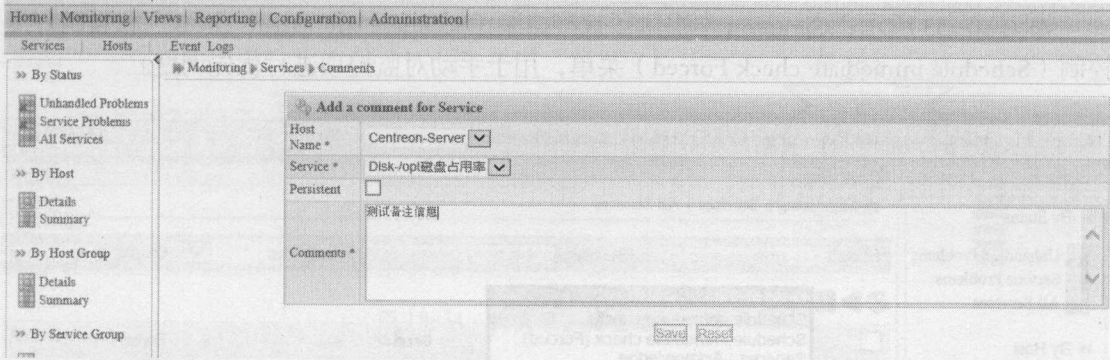


图 10-31 备注添加界面

在添加备注信息的时候，选择 Persistent 项，可以使 Nagios 在重启之后，依然保留之前添加的各类备注信息，而不是丢失。

在 Centreon 中，备注的管理界面在下列路径中：Monitoring→Services，接着单击左侧竖状菜单里的 Comments 按钮，可进入备注管理界面，在这里，您可以管理所有服务的备注信息，如图 10-32 所示。

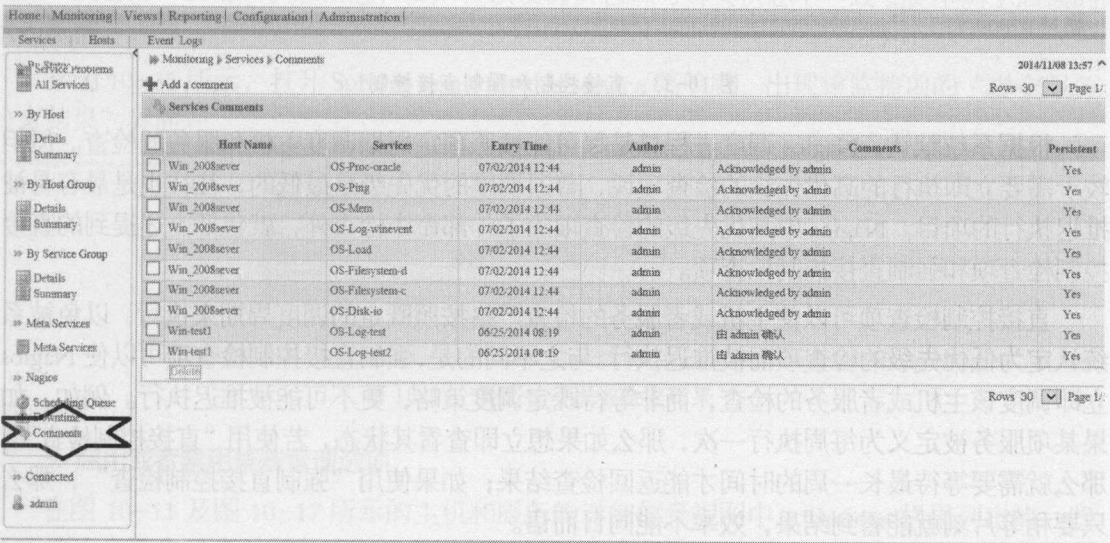


图 10-32 管理备注列表

10.8.5 对于调度任务的直接控制

相较于 Nagios 调度进程对于主机和服务检查任务的自动的、按计划的调度，Centreon 也提供了对于监控对象的计划外直接调度——即以手动形式直接控制调度任务，例如立即发起一个或多个检查。这样做的目的是能够在计划调度之前立即观察到监控项的状态，而不必等待可能需要几分钟的例行调度检查结果。

1. 直接控制和强制直接控制

如图 10-33 所示，Centreon 中提供了直接控制（Schedule immediate check）和强制直接控制（Schedule immediate check Forced）菜单，用于手动对监控项进行优先调度。

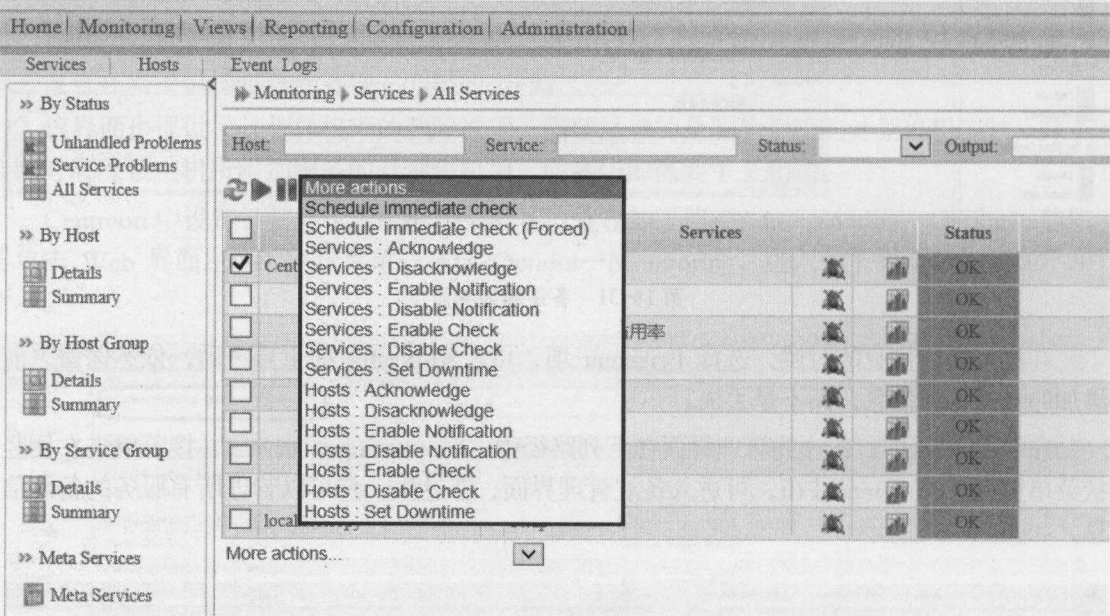


图 10-33 直接控制和强制直接控制

根据系统默认，Nagios 调度进程总是尽可能按照预定周期调度主机和服务的检查。但相较于需要立即执行的高优先级的检查项说，普通调度的优先级是最低的，往往也是最容易被推迟执行的项目。Nagios 中高优先级的检查项除了外部命令检查外，就包括上述提到的直接控制检查项和强制直接控制检查项。

直接控制检查项可以使主机或者服务的检查严格按照既定的调度周期来执行，以免被系统认定为低优先级的检查项而被推迟执行。与之不同的是，强制直接控制检查项可以使 Nagios 立即调度该主机或者服务的检查，而非等待既定调度策略，更不可能被推迟执行。例如，如果某项服务被定义为每周执行一次，那么如果想立即查看其状态，若使用“直接控制检查”，那么就需要等待最长一周的时间才能返回检查结果；如果使用“强制直接控制检查”，那么只要稍等片刻就能看到结果，效率不能同日而语。

当用户增加了一些服务监控项，重启 Nagios 调度进程后，需要立即查看这些监控项的输出以及返回状态，那么可以对这些服务监控项执行“强制直接控制”检查。

Centreon 同时提供了这两项直接控制菜单，在实际运行过程中，往往使用“强制直接控制检查”命令更多一些。

2. 提交服务检查结果

如图 10-34 所示，在服务的详细信息页面中，我们可以找到 Submit result for this service 命令选项，前提是必须开启该服务的被动检查模式，即使 Passive Checks 项的状态为 Enabled。

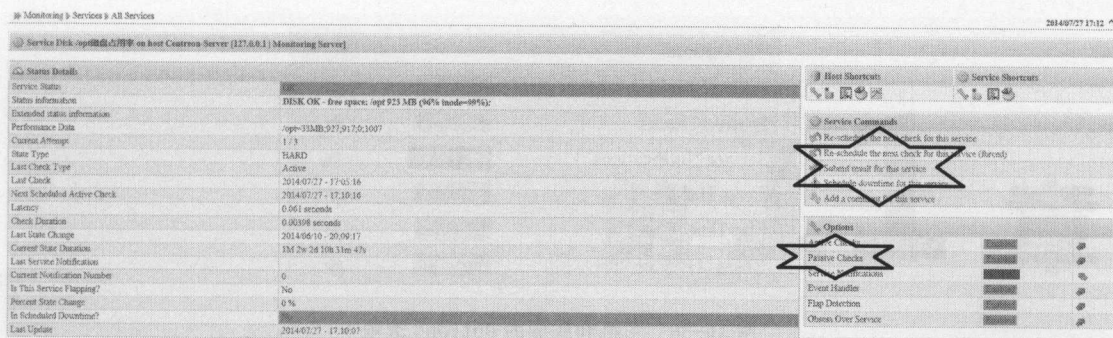


图 10-34 提交服务检查结果

在 Centreon 和 Nagios 中常用的主动监控模式相比，被动监控模式更加依赖于监控项自身的告警策略。在绝大多数场景中，采取被动监控模式的监控项并不会发送所有状态，而是仅发送告警信息，并不会发送从告警状态到正常状态后的恢复信息。此特性往往不利于管理人员掌握实时的监控项状态信息，因此在被动监控模式下，当问题监控项恢复正常后，手动提交该监控项的服务检查结果是很有必要的。

例如，某台网络设备发送了一条 SNMP trap 告警信息，管理人员接收到了告警并修复了设备，但是设备并不会自动发送一条状态为“正常”的恢复信息。这就需要管理人员手工提交该设备的“正常”状态信息，否则，该设备状态就与真实状态不一致了，不利于后续的状态监控。

如图 10-35 所示，打开 Submit result for this service 菜单后，出现该监控项的“提交服务检查结果”界面。

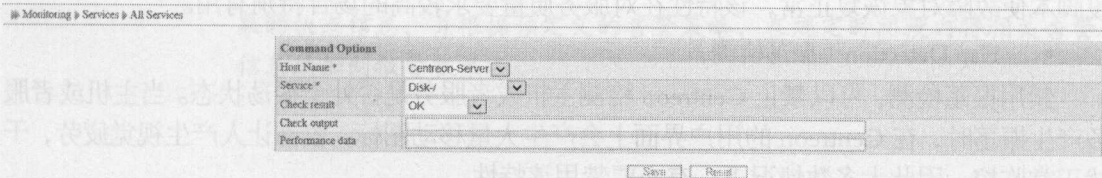


图 10-35 提交被动服务检查结果

3. 启用或者禁用某些管理选项

在图 10-11 及图 10-17 所示的主机和服务的详细信息视图中，Options 栏里列出的一些选项默认为开启或者关闭状态。这些选项的默认值在主机和服务的模板里都有配置。通过 Web 界面改变这些选项只能起到“运行时生效”的作用，当 Nagios 调度进程重启，或者服务器重启之后，用户在 Web 界面中对于这些选项所做的更改便会丢失。

具体来说，当用户单击这些选项后的“箭头”图标来更改这些选项时，Centreon 会向后台运行的 Nagios 等调度进程直接发送命令，更改调度进程中这些选项的配置，并立即生效。而不是像修改其他配置项一样，需要导出并生成 Nagios 调度进程配置文件，并重启 Nagios 才能生效，如图 10-36 所示。







Options		
Active Checks	Enabled	
Passive Checks	Enabled	
Service Notifications	Disabled	
Event Handler	Enabled	
Flap Detection	Enabled	
Obsess Over Service	Enabled	

图 10-36 主机和服务的 Options 选项

▪ Active Checks 和 Passive Checks（主动检查与被动检查）

前面提过，主动检查是指 Centreon 主动发起对于服务（监控项）的检测，既是一种主动行为，也是最常规的一种检测行为。启用对于监控项的主动检查，需要为该监控项配置并启用一个探针才能进行。一般来说，禁用主动检查在日常管理中并非好用，更常用的办法是禁用该服务项，或者为该服务项设置停机时间。

▪ Service Notifications（服务通知消息）

使用该选项，可以禁用或者启用服务的通知消息。当出于某种考虑，管理人员不再希望接收服务通知消息时，可以选择禁用该服务的通知消息。在主机的相关菜单中，可以选择启用或者禁用该主机所关联的所有服务的通知消息，这在该主机宕机或者检测探针失效时有很大用途，可以很大程度上降低因主机宕机而产生的服务故障通知消息的数量。

▪ Event Handler（事件管理）

该选项允许管理员在主机或者服务产生故障告警的情况下，执行一个命令、或者运行一段脚本使故障对象恢复正常，该特性在对服务质量要求较高的场合特别有用。

▪ Flap Detection（振荡检测）

禁用振荡检测，可以禁止 Centreon 检测主机或者服务是否处于振荡状态。当主机或者服务产生振荡时，在 Centreon 的用户界面上会产生大量移动图标，容易让人产生视觉疲劳，干扰正常监控，因此大多数情况下，有必要禁用该特性。

▪ Obsess Over Service（服务纠缠）

该选项决定了 Nagios 是否对某项服务的多个检测结果进行“纠缠”，以执行通过 `obsessive compulsive service processor command` 选项定义的命令。正如 9.2.2 小节所述，该选项在分布式的 Nagios 监控模式下有效，在集中监控模式下，可以禁用该选项。

第 11 章

Centreon 的配置

有了前面的介绍，我们此时已经了解到，在主动模式下，Centreon 的主要任务就是发起对主机的状态，以及所关联服务状态的检测，搜集并存储主机和服务状态数据和性能数据。这些检测任务通常是由部署在 Centreon 中心监控服务器端的检测探针和被检测端的检测插件来执行的。通过为检测探针或者插件配置合适的参数，例如被检测对象的 IP 地址、主机名、监控阈值等参数，可以实时探测并根据阈值判断被检测对象的状态信息，并根据预定义的告警策略，在规定时间内将相关告警信息通知到报警联系人。

2. 删除监控对象

1) 在监控对象列表中，选择要删除的对象：

2) 选择下拉列表“More actions”中的“Delete”（删除）。

11.1 Centreon 的监控对象模型

1. 数据模型

Centreon 的数据模型直接脱胎于 Nagios，这也是经许可的，如图 11-1 所示。

在 Centreon 中，可以定义主机模型与服务模型，具体来说就是定义主机模板和服务模板，存储在数据库中。在实际部署主机和服务的时候，直接继承各自的模板，可以快速生成并批量部署主机和服务，而且采用模板管理也有利于批量调整主机和服务的参数，可谓一举多得。同时，主机模型和服务模型还与联系人及

联系人组存在关联关系，当发生故障告警后，可直接通知到主机和服务对应的联系人或者联系人组。而联系人组的引入又便于系统管理员灵活调整故障告警的通知对象，因此整个数据模型显得灵活可变。

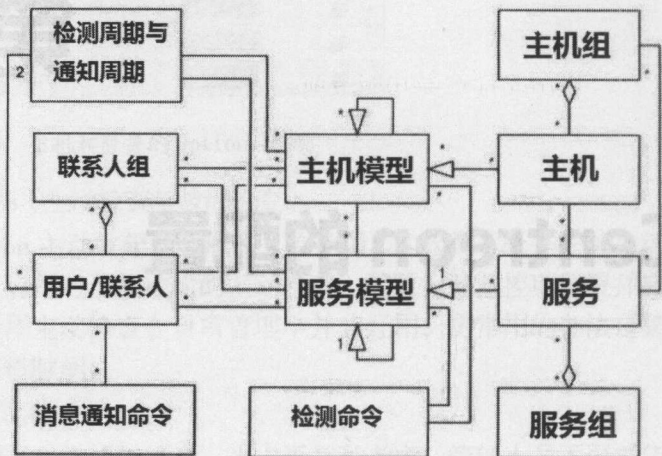


图 11-1 Centreon 的数据模型

2. 配置顺序

Centreon 中对对象模型的关联关系决定了在实际进行配置时的先后顺序，一般按照关联关系的数量从少到多开始配置。先配置关联关系最简单的对象，然后再其上堆砌更多对象。一般说来，遵循如下配置步骤，可使一套监控方案快速落地。

- （1）配置检测命令和检测周期，因为这些对象是最频繁也最容易被复用的——多个检测对象可以调用同一条检测命令，执行同一种检测周期。
- （2）配置用户组和联系人组，以便于后续监控对象的关联。
- （3）配置主机模板和服务模板，便于后续批量生成主机和服务。
- （4）根据模板配置主机以及与之关联的各类监控服务项，然后再根据每项主机和服务的特性进行灵活微调。
- （5）配置主机组和服务组，并将其与用户组和联系人组关联起来。

11.2 通用功能配置界面

1. 启用和禁用监控对象

Centreon 中的每一类监控对象都可以被临时禁用，目的是多样的，例如，在使某台主机或者服务项生效之前进行预先配置、准备一套主机或者服务模板以便后续批量配置等等。与

已经启用并生效的监控对象不同的是，被禁用的监控对象的定义不会被导出到 Nagios 调度进程的配置文件中去，因此我们必须重视 Centreon 中存储对象的启用状态，以及 Nagios 配置文件中对象启用状态的一致性。

有如下两种方法来启用或者禁用监控对象：

(1)使用图标来启用或者禁用监控对象，例如，单击图 11-2 中的箭头图标，可以分别启用或者禁用服务项。

More actions... <div><div></div>Add</div>					Rows 30	Page 1/1
<input type="checkbox"/>	Host	Service	Scheduling	Parent Template	Status	Options
<input type="checkbox"/>	Centreon-Server	Disk-/	5 min / 1 min	-> SNMP-DISK-/ -> generic-service	Disabled	1
<input type="checkbox"/>		Disk-/home	5 min / 1 min	-> SNMP-DISK-/home -> generic-service	Enabled	1
<input type="checkbox"/>		Disk-/opt磁盘占用率	5 min / 1 min	-> SNMP-DISK-/opt -> generic-service	Enabled	1
<input type="checkbox"/>		Disk-/usr	5 min / 1 min	-> SNMP-DISK-/usr -> generic-service	Enabled	1

图 11-2 单击箭头启用和禁用监控对象

(2) 通过单击菜单 Configuration → Services，进入服务的配置页面，在 Service Extended Info 选项卡的 Additional Information 项中，可以选择启用或者禁用对象，如图 11-3 所示。

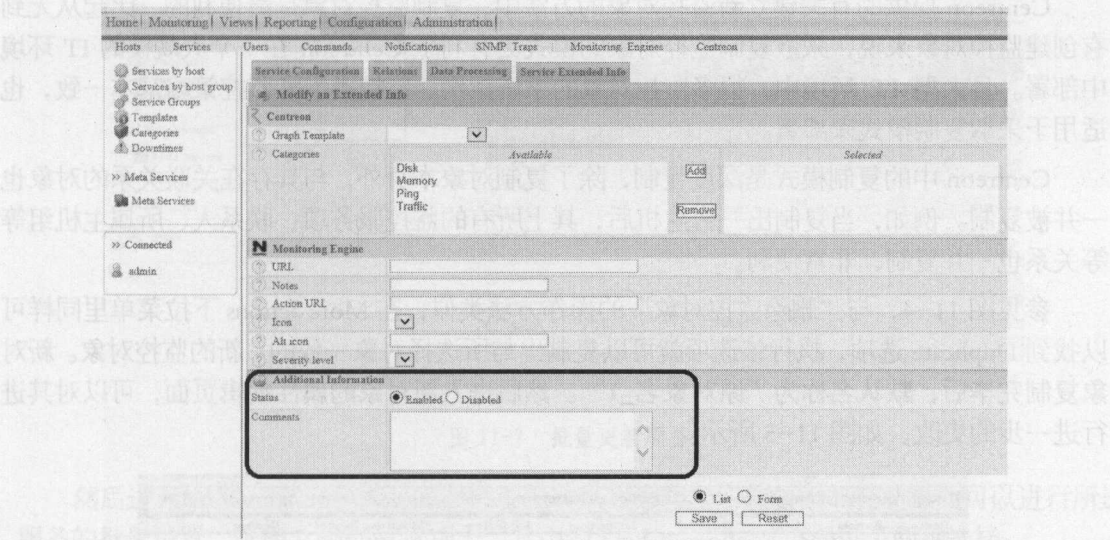


图 11-3 选择菜单启用或者禁用对象

2. 删除监控对象

对于监控对象的删除动作在 Centreon 中是不可逆的，因此应该慎重操作。因此对于一般的监控项配置操作来说，禁用该监控对象比删除该对象更有利于以后的恢复。删除监控对象可遵循下列步骤：

- (1) 在监控对象列表中，选择要删除的对象；
- (2) 选择下拉列表“More actions”中的“Delete（删除）”选项；
- (3) 在弹出的删除确认对话框中单击“确认”。

如图 11-4 所示。

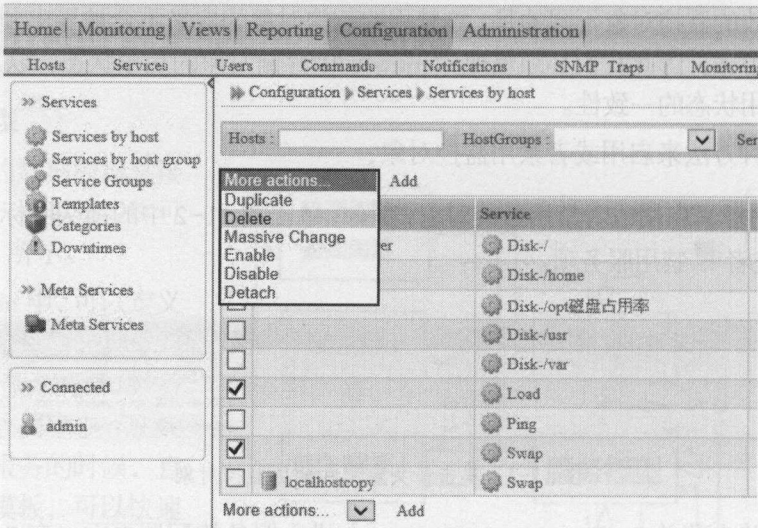


图 11-4 删除监控对象

3. 复制监控对象

Centreon 提供的有关建立新监控对象的方法中，复制监控对象是最便利的。比起从无到有创建监控对象来说，采取复制监控对象的模式更有利于 Centreon 在一个大规模的 IT 环境中部署。且大型 IT 环境中，很多监控对象的系统架构、监控类型和监控策略基本一致，也适用于采取复制模式来部署。

Centreon 中的复制模式是深度复制，除了复制对象本身外，与其存在关联关系的对象也一并被复制。例如，当复制出一台主机后，其上所有的监控服务项、联系人、所属主机组等等关系也一并复制，非常便利。

参见图 11-4，与“删除监控对象”的操作方法类似，在 More actions 下拉菜单里同样可以找到 Duplicate 选项，执行该选项就可以复制出与所选择对象一致的、新的监控对象。新对象复制完毕后，默认名称为“原对象名_1”。然后进入新对象的属性编辑页面，可以对其进行进一步的更改，如图 11-5 所示。

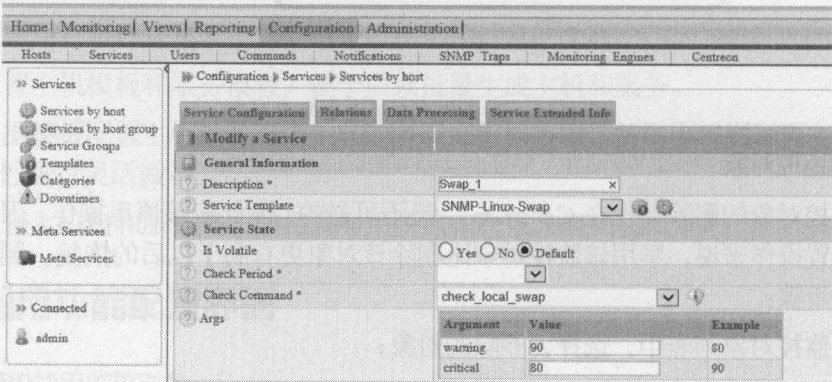


图 11-5 复制出来的监控对象

如图 11-6 所示为执行多次复制后的结果：

<input type="checkbox"/>	localhostcopy	Swap	5 min / 1 min	
<input checked="" type="checkbox"/>		Swap_1	5 min / 1 min	
<input checked="" type="checkbox"/>		Swap_1_1	5 min / 1 min	
<input type="checkbox"/>		Swap_1_1_1	5 min / 1 min	

More actions

图 11-6 多次复制后的对象列表

4. 批量设置或者批量更新对象属性

批量设置是指一次性地改变多个监控对象的一个或者多个属性。批量设置在更新多个同类型监控对象的时候特别有用。例如，当使用监控对象的“复制”功能复制出多个服务项之后，想要批量改变这些服务项的所属主机，而非逐个更改，就可以使用“批量更新”功能，一次性地更新这些复制服务的关联主机，由旧的主机关联至新的主机，从而实现服务的批量调整。

如图 11-7 所示，选择复制出的服务，单击 More actions 菜单里的 Massive change 选项：

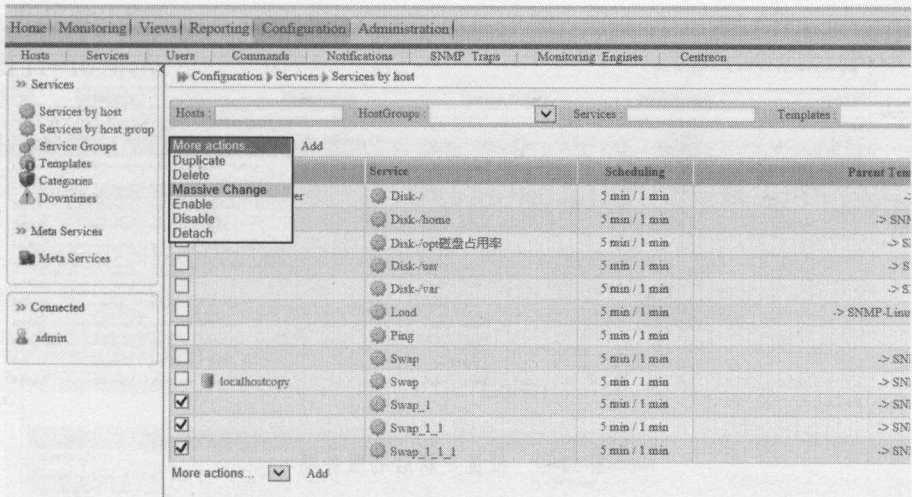


图 11-7 批量更新服务

然后进入 Configuration→Services→Services by host 菜单所在页面，在此页面可以进行所选服务的批量设置。选择 Relations 选项卡，其中的 Update mode 选项提供了两种选择：

- Incremental（增量）模式，即以增量模式进行服务属性的批量设置，是指在所要更改的多个监控对象属性列表上增加这些属性。
- Replacement（替换）模式，即以替换模式进行服务属性的批量设置，是用所选属性列表来替换所要更改的监控对象上的原有属性列表。

如图 11-8 所示，如果我们想要使所批量更新的监控服务项既从属于 Centreon-Server 主机，又从属于 localhostcopy 主机，可以选择 Incremental 模式。在此，我们是想让这些批量更新项由从属于 localhostcopy 主机变更为从属于 Centreon-Server 主机，因此我们选择 Replacement 模式。

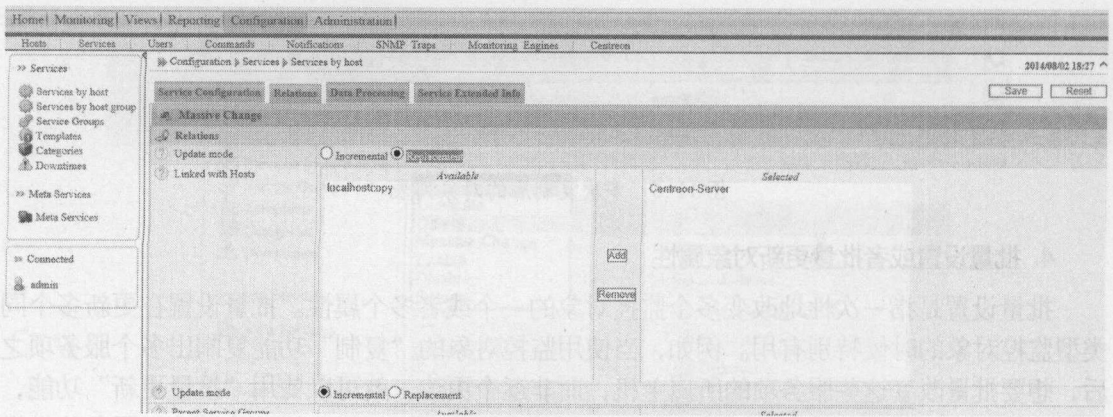


图 11-8 批量更新中的增加模式与替换模式

单击 Save 保存后，我们可以看到如图 11-9 所示的，复制出的这些服务已经从属于 Centreon-Server 主机了。

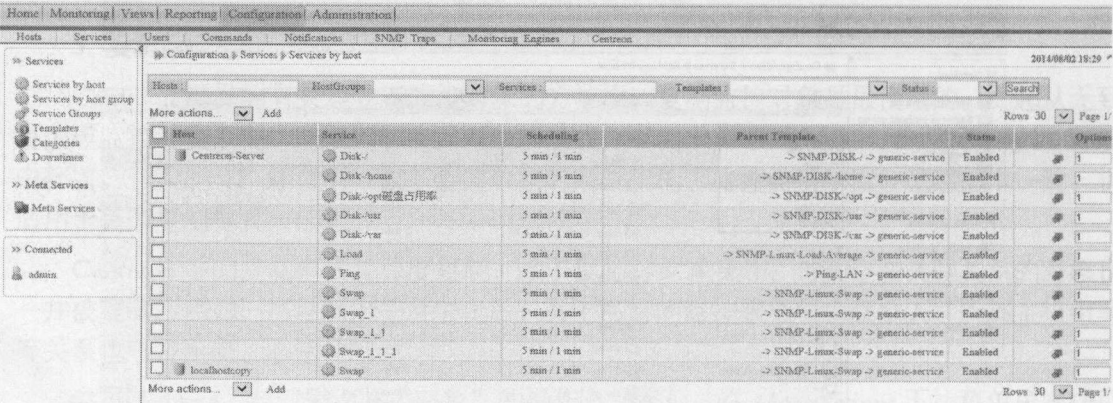


图 11-9 批量更新后的服务项

一般来说，Incremental 模式在调整主机和主机组关系时用的比较多。例如，可以新建主机组，然后选择一些想要加入该主机组的主机，使用 Incremental 模式将所选择主机批量更新至新建主机组中。如此一来，在不改变这些主机原有属组的情况下，又定义了新的主机组，很大程度上方便了主机之间关系的建立。

11.3 Nagios 配置文件的生成与部署

通过学习 Centreon 的架构，我们已经了解到，绝大多数通过 Centreon 的 Web 用户界面所做的配置修改都存放在后台 MySQL 数据库中，这些配置在被 Nagios 等调度进程识别之前，是不会生效的。要想使 Nagios 识别并加载修改后的配置，需要遵循下列 4 个步骤：

- (1) 使用 Centreon 生成符合 Nagios 格式的配置文件；
- (2) 生成的配置文件能够被 Nagios 校验通过；
- (3) 将生成的配置文件移动到合适的目录，可以通过移动文件或者 SSH 复制的方式；

(4) 使 Nagios 能够重新加载配置文件。

以上步骤可以在菜单 Configuration → Monitoring Engines → Generate 中执行。

1. 生成并校验 Nagios 配置文件

在执行完主机/服务项的增删改后, 若想使其在 Nagios 中生效, 必须将修改完毕后的配置项导出为 Nagios 配置文件, 并重启 Nagios 调度进程。

按照默认配置, Centreon 的 Web 用户界面产生的 Nagios 配置文件位于 Centreon 监控服务器的 “/usr/local/centreon/filesGeneration/nagiosCFG/x” 路径下, 其中的 x 代表调度进程的序号, 一般是 1。在创建配置文件之前, Centreon 必须调用 “/usr/local/nagios/bin/nagios -v” 命令来校验配置文件是否合法, 似乎能够被 Nagios 调度进程正确识别。

为达到上述目的, 必须采取以下步骤:

(1) 进入菜单 Configuration → Monitoring Engines → Generate, 选择 Generate Configuration Files 项和 Run monitoring engine debug (-v) 项, 然后单击 Export 执行配置文件生成和校验工作。

如图 11-10 所示。

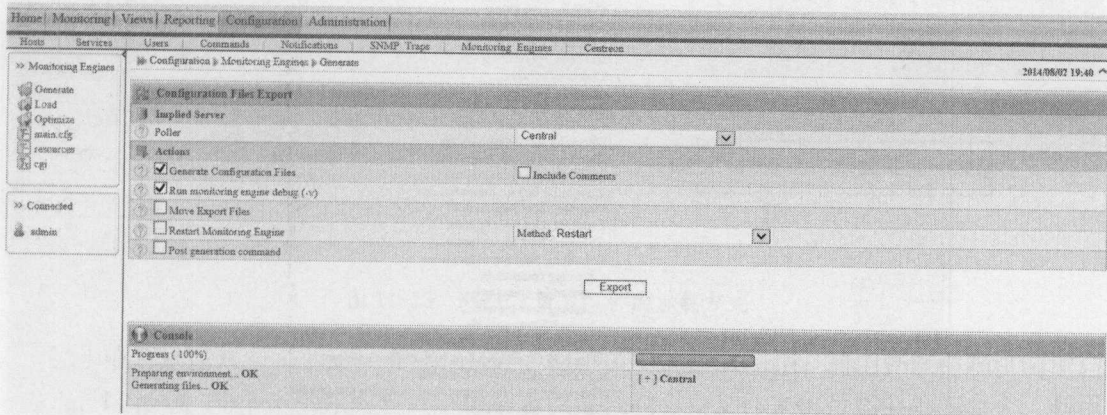


图 11-10 生成并校验 Nagios 配置文件

(2) 如果生成的 Nagios 配置文件不合法, 单击 “+” 会显示出具体的警告或者错误原因, 如图 11-11 所示, 显示出警告原因为 localhostcopy_1 主机上不存在监控服务项, 该警告是可以忽略的。

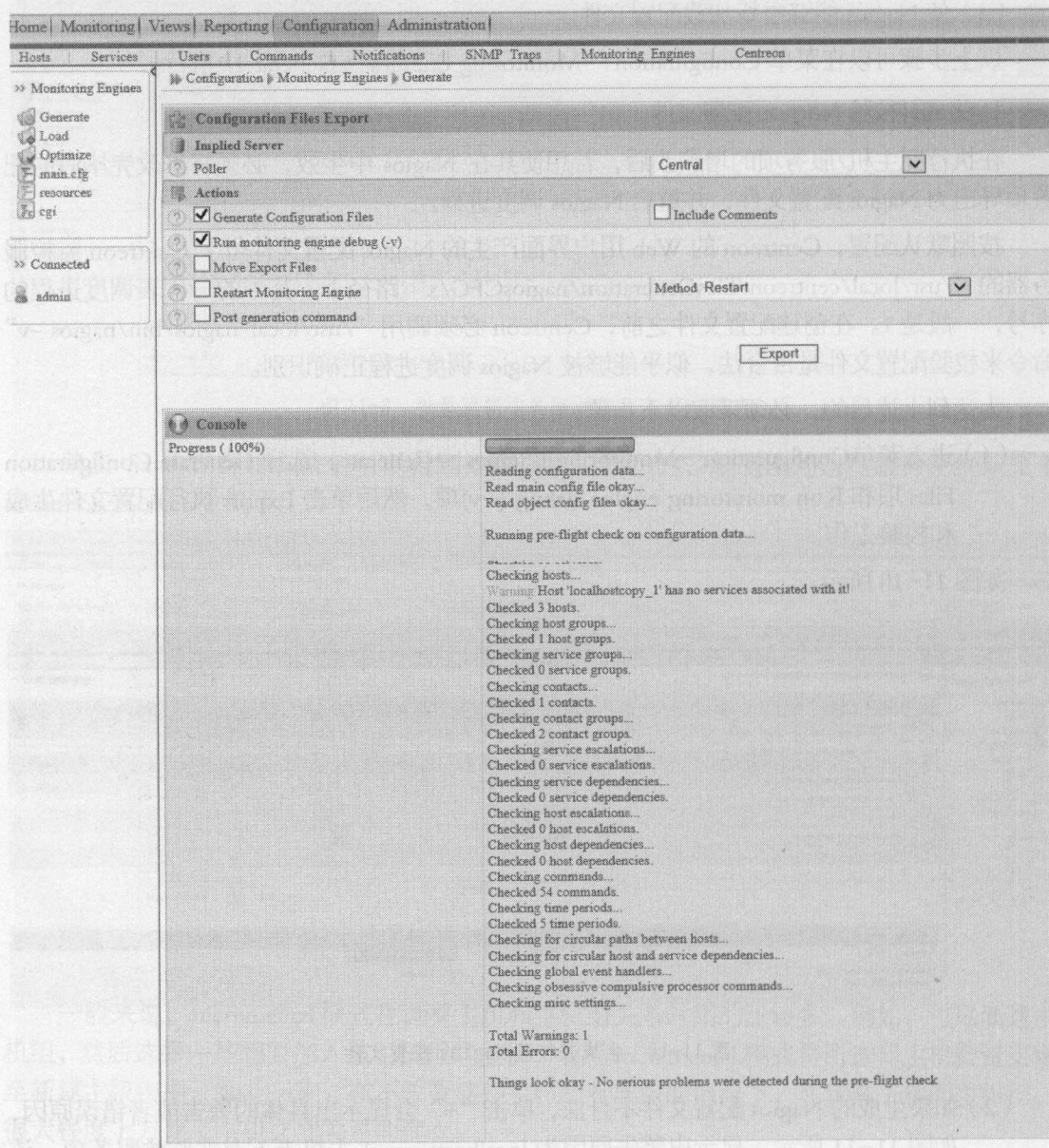


图 11-11 校验 Nagios 配置文件产生警告信息

2. 移动并重新加载 Nagios 配置文件

Centreon 生成 Nagios 调度进程的配置文件之后，会立即将配置文件存放到后者的指定目录中，供 Nagios 加载。如果 Centreon 服务器和 Nagios 调度进程在同一台服务器上，只需要在文件目录间移动这些配置文件即可。如果是采用分布式架构，Centreon 和 Nagios 位于不同的服务器上，就需要 Centreon 通过 SSH 连接的方式将配置文件复制到远端服务器的指定目录。以上工作执行完毕之后，Centreon 会通知 Nagios 调度进程重新加载更新完毕的配置文件，有以下两种方式：

(1) 直接在 Nagios 所在服务器上重启 Nagios 调度进程。在 Linux 操作系统中，一般使用 service nagios restart 命令来重启，如图 11-12 所示。

```
[root@monitor bin]# service nagios restart
Running configuration check... done.
Stopping nagios: done.
Starting nagios: done.
```

图 11-12 重启 Nagios 服务

或者使用 “kill 进程 id” 命令直接杀掉原有进程，再重启 Nagios 进程的方式也可以。

(2) 在 Restart Nagios 的 Method 下拉菜单中选择 Reload，通知 Nagios 重新加载配置文件。比起重启的方式，重新加载的方式更加安全快捷。一般来说，该方式是最普遍采用的方式。

如图 11-13 所示。

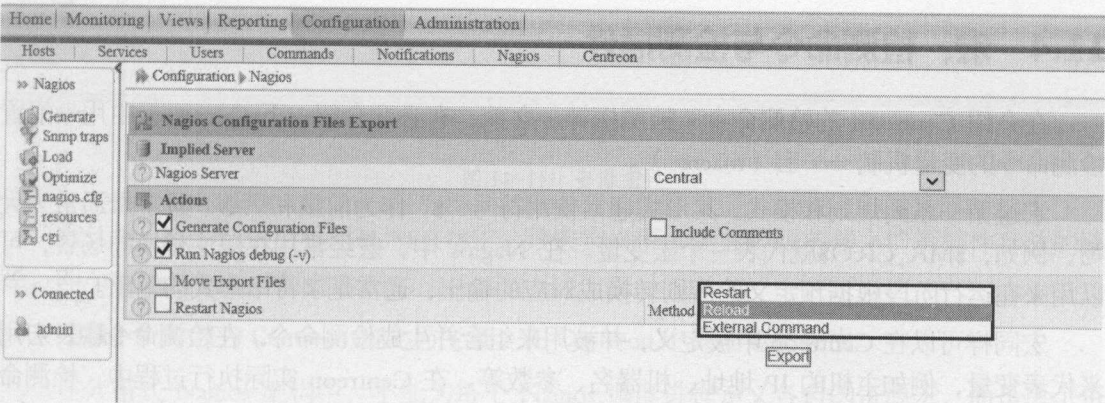


图 11-13 Nagios 配置文件的加载方式

(3) 或者是在 Restart Nagios 的 Method 下拉菜单中选择 Restart，利用外部控制机制重启 Nagios。Nagios 会定时轮询本地目录中存放的命令文件来执行相关外部命令，利用该机制，Centreon 会将 Nagios 重启命令以文件的形式存放到 Nagios 的目录中，后者一旦发现该文件，会立即执行自我重启操作。

以上 3 种重新加载 Nagios 配置文件的方式相比，第 2 种更为安全快捷，且不会干扰正常的 Nagios 进程运行(在 Linux 群集环境中，Nagios 进程一般作为群集资源组存在，重启 Nagios 进程可能导致资源组发生切换，因此最好不要频繁重启)，因此更加可行。

要执行上述重启步骤，只需要在图 11-11 的基础上，再选择 Move Export Files、Restart Monitoring Engine 和 Post generation command 选项，再次执行 Export 操作即可。注意，在 Restart Monitoring Engine 后，您可以选择上述 3 种重新加载 Nagios 配置文件的方式，在图 11-14 中，我们选择以 Reload 的方式重新加载 Nagios 配置文件。

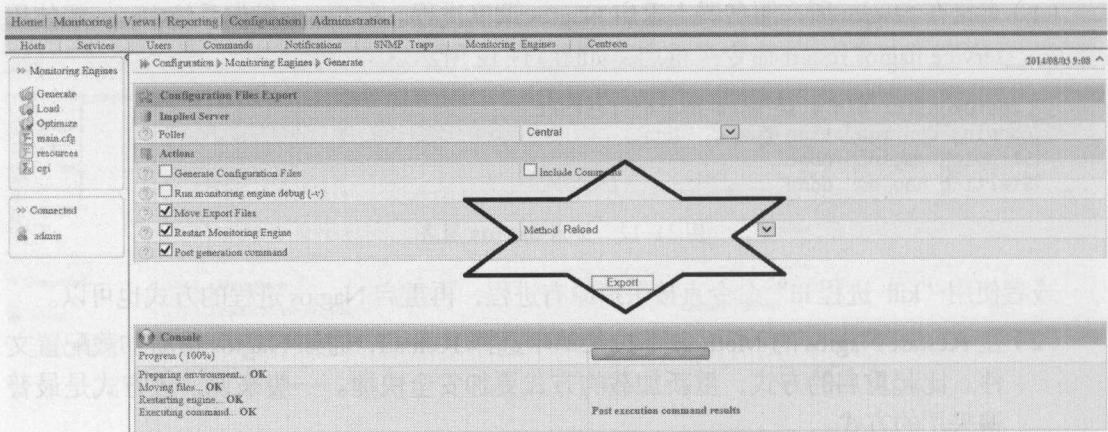


图 11-14 选择以 Reload 方式重启 Nagios

11.4 宏、检测命令与检测插件

在介绍 Centreon 的检测命令（检测探针）之前，有必要介绍一下 Centreon 中用来配置检测命令的变量机制——宏（macros）。

宏遵循一系列规则和模式，其形式通常使用符号“\$”作为前缀和后缀，且字母通常为大写，例如，\$MA_CRO\$就代表一个宏变量。在 Nagios 中，宏经常用在语法替换的场景，可以用来在运行阶段根据预定义的规则转换成对应的输出，通常是字符串形式的输出。

宏同样可以在 Centreon 中被定义，并被用来组合并生成检测命令。在检测命令中，宏用来代表变量，例如主机的 IP 地址、机器名、参数等。在 Centreon 实际执行过程中，检测命令中的宏会被替换成一系列变量，并生成实际的、可执行的检测命令。

在 Centreon 中，主要用到了以下 5 种宏：

- 标准宏。
- 资源宏。
- 自定义宏。
- 按需而生的宏（Demand Macro）。
- 参数宏。

Centreon 中宏的概念直接来源于 Nagios，相关文档在 Nagios 中也能找到。

1. 标准宏

标准宏是 Centreon 系统中默认的宏类型。每一个标准宏都有各自的专有用途和所替代的变量。Centreon 已经提供了用于消息通知、主机名和服务名定义、主机和服务检查、事件处理等专有用途的宏，这些宏是系统预定义的，不能被随意更改。

标准宏的用途通常从其定义就能识别出来，例如，\$HOSTNAME\$宏用于替换主机名，\$SERVICESTATE\$宏用于代表检测项的状态，而\$CONTACTEMAIL\$代表通知联系人的邮箱等。某些标准宏还具备统计意义，例如\$TOTALHOSTUP\$。

提示：所有 Centreon 中使用的标准宏的定义都可以在官方的 Nagios 文档中找到，可以访问如下网址了解详情：http://nagios.sourceforge.net/docs/3_0/macrolist.html。

2. 资源宏

资源宏在 Centreon 中是全局变量，意味着可以被任意检测命令所引用。资源宏遵循 \$USERn\$ 的命名规则，其中“n”为 1 到 256 的数字，这也限制了资源宏最多只能有 256 个。

可以通过如下方式列出系统中已经定义的资源宏，单击菜单 Configuration→Monitoring Engines，在左侧的竖状菜单上单击 resources，如图 11-15 所示。

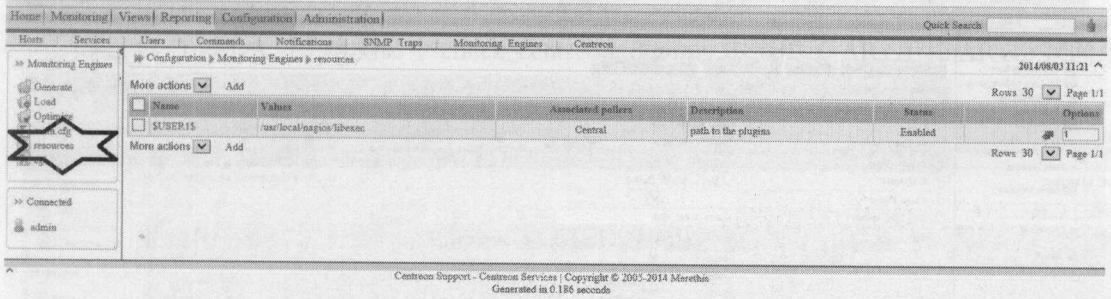


图 11-15 资源宏

默认地，Centreon 会提供 \$USER1\$ 宏，该宏定义了中央监控服务器上检测插件存放的路径，即 /usr/local/nagios/libexec。

注意：资源宏为 Centreon 的全局变量，在 Centreon 中只能定义一次。

在分布式 Nagios 架构中，不能为每个 Nagios 调度进程定义各自的资源宏，而是单个资源宏可以被多个 Nagios 调度进程所使用，例如 \$USER1\$ 资源宏，在多个 Nagios 调度进程中都代表 /usr/local/nagios/libexec 目录。

如图 11-16 所示，是资源宏的定义页面。单个资源宏可与多个 Nagios 监控实例相关联。

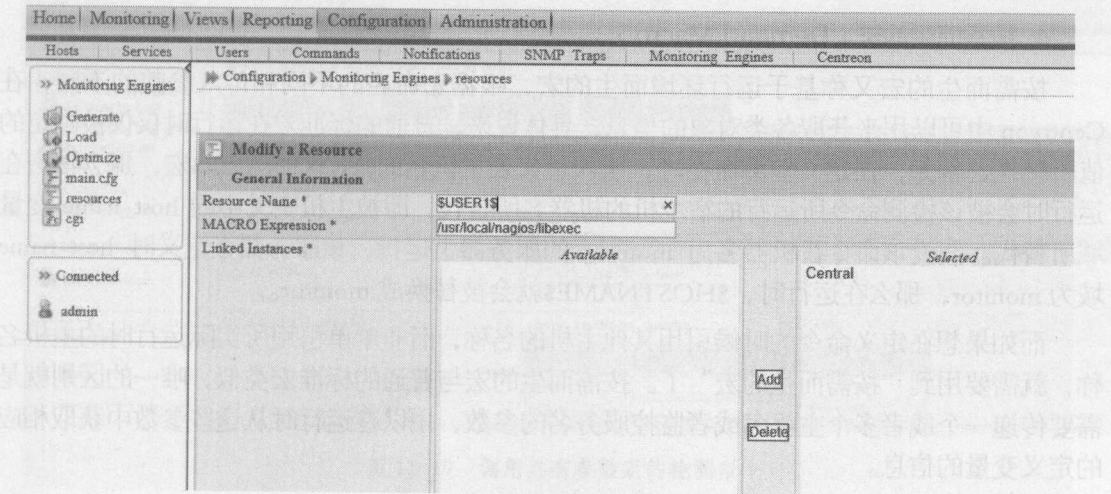


图 11-16 资源宏的定义页面

3. 自定义宏

顾名思义，自定义宏是 Centreon 用户自行定义的，用于主机和服务模板创建、监控项定义、检测命令定义的一系列变量。一般来说，与主机有关的宏通常为 `$_HOST$` 类型，而与服务检测项有关的宏通常为 `$_SERVICE$` 类型。

在 Centreon 中的“主机定义与配置页面”和“服务定义与配置页面”中，都可以自由定义自定义宏。进入菜单 Configuration → Services，单击服务名称，进入服务定义与编辑页面，如图 11-17 所示。

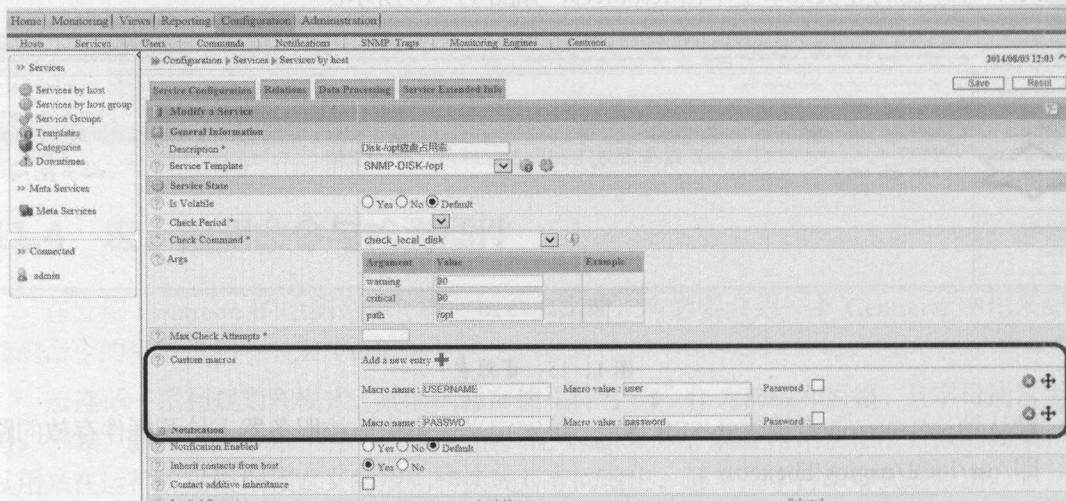


图 11-17 自定义宏

如图 11-17 所示，单击“+”符号可以增加自定义宏。由于是自定义宏，因此宏的名称不强制使用“\$”作为前缀和后缀，只要便于理解就可以。在本例中，我们为该服务定义了 USERNAME 和 PASSWD 两个宏，那么在该服务实际被调用过程中，以上两个宏会分别被 user 和 password 所代替。

4. 按需而生的宏 (Demand Macro)

按需而生的宏又称基于运行环境而生的宏，根据宏命名的不同和传入参数的不同，在 Centreon 中可以用来获取各类对象的信息，具体说来，普通的标准宏在运行时仅仅被固定的值所替代，例如，在定义检测命令时，引入了代表主机名的 `$HOSTNAME$` 宏，那么该宏在运行时会被该检测命令所运行的宿主机的机器名所替代，即被主机定义中的 `host_name` 变量域所替代。假设该命令在机器名为 `monitor` 的服务器上运行，该服务器在定义时 `host_name` 域为 `monitor`，那么在运行时，`$HOSTNAME$` 就会被替换成 `monitor`。

而如果想在定义命令的时候引用其他主机的名称，而非单单引用所实际运行时的主机名称，就需要用到“按需而生的宏”了。按需而生的宏与普通的标准宏类似，唯一的区别就是需要传递一个或者多个主机名或者监控服务名的参数，用以在运行时从这些参数中获取相应的定义变量的信息。

按需而生的宏与标准宏一样，都是从 Nagios 系统继承而来，其命名可以根据如下规则：

```
$MACRONAME:<parameter 1>[:<parameter 2>...]$
```

上面的\$MACRONAME 字符串就是普通的标准宏，唯一不同的是它带上了参数，意味着可以获取别的主机上所定义的变量信息。实际上，几乎所有的标准宏都可以被看成是需求宏，唯一的区别就是使用的时候需要提供至少一个参数。

\$HOSTSTATE:server1\$宏可以被监控服务器 server1 的主机状态所替代
\$CONTACEEMAIL:frank\$可以被系统管理员 frank 的电子邮箱地址所替代。

5. 参数宏

参数宏常用于检测命令定义中，用来代表检测命令所需的参数。其形式为\$ARGn\$，其中 n 为从 1 开始，随参数数量增加而递增的数字。

如图 11-18 所示，在检测命令 check_centreon_dummy 中提供了 ARG1 和 ARG2 共两个参数：

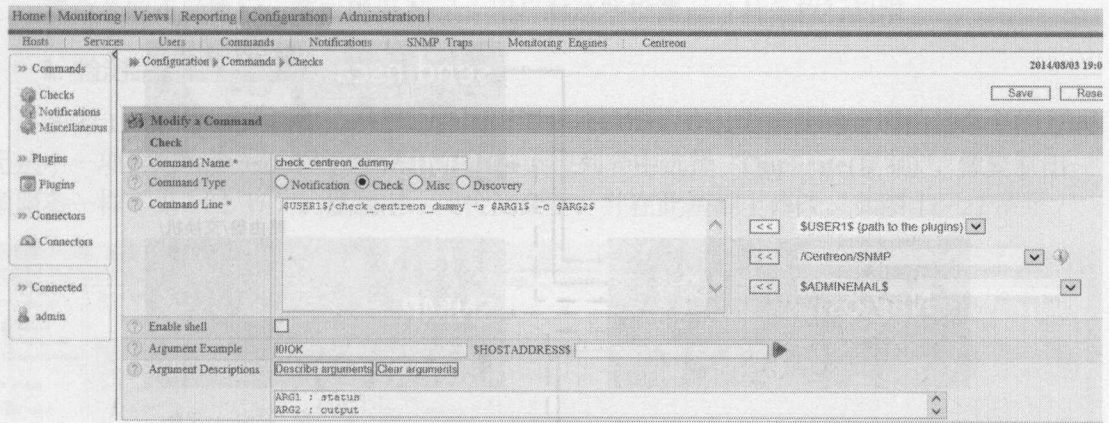


图 11-18 参数宏

那么在定义服务器并调用 check_centreon_dummy 检测命令时，必须提供两个参数，如图 11-19 所示。

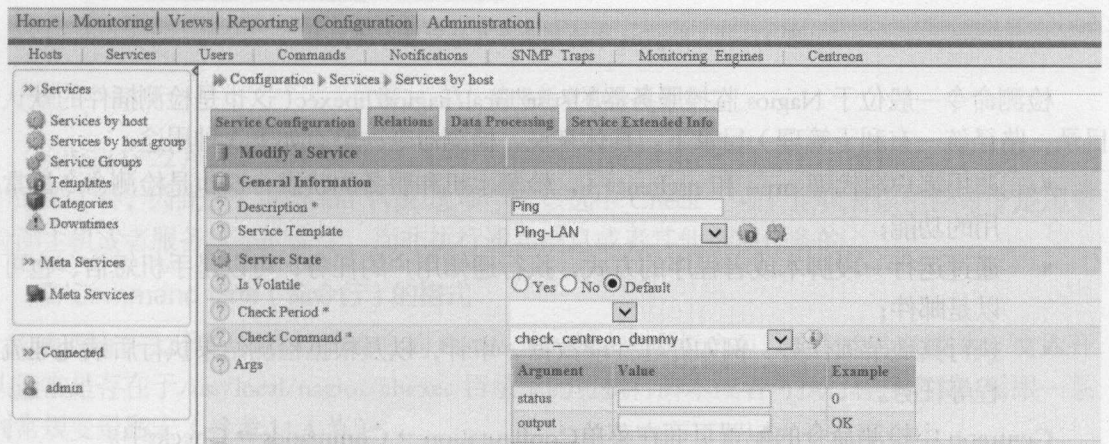


图 11-19 调用具有参数宏的检测命令

11.5 检测命令与检测插件

检测命令, 又称检测探针、或者探测器 (Probe), 是指配置在 Centreon 监控服务器上的可执行脚本或者程序, 用以执行对被监控主机或者服务的检测、或者执行通知消息等任务。而检测插件大多数是位于被检测主机端的可执行脚本或者程序, 可以被检测命令调用, 用以承担具体的检测任务。当然并不是所有检测插件都位于被监控端, 某些可以执行远程访问的插件也可以部署在 Centreon 监控服务器上。

如图 11-20 所示, 位于 Linux/Unix 操作系统平台和 Windows 平台的同一款 check_log 日志检测插件, 被各自平台的 nrpe 和 nsclient++ 等代理进程驱动 (参照 7.2.2 小节相关进程作用的解释), 获取日志告警信息和性能数据, 并返回给位于 Nagios 监控服务器端的检测命令, 例如 check_nrpe。

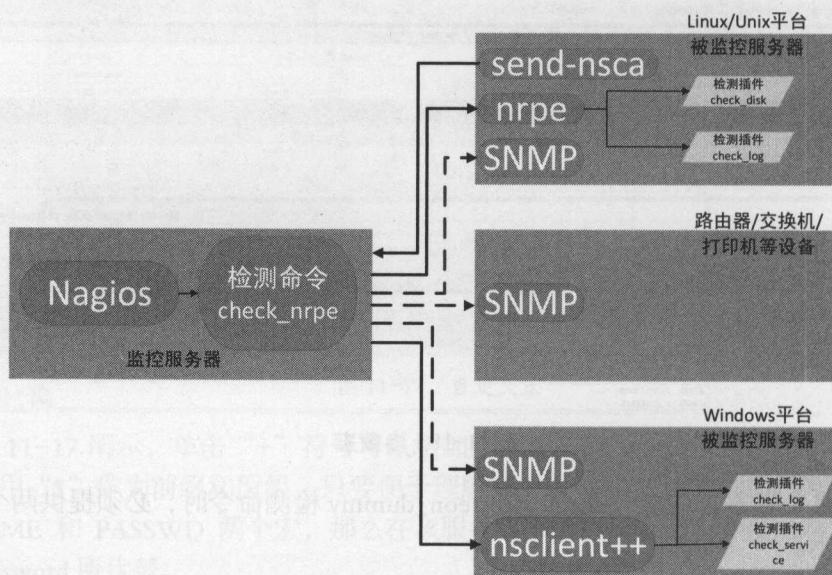


图 11-20 Nagios 中检测命令与检测插件的关系

检测命令一般位于 Nagios 监控服务器的 /usr/local/nagios/libexec (这也是检测插件的默认目录, 路径统一有利于管理) 目录下。具体来说, 检测命令一般有如下 3 种用途:

- 调用客户端代理 nrpe 和 nsclient++, 检测主机和服务的状态, 这也是检测命令最常用的功能;
- 通过运行一段脚本或者程序的方式, 执行通知用户的任务, 可以是手机短信、也可以是邮件;
- 执行其他杂项任务, 例如处理性能数据、审计, 以及根据检测结果执行后续处理流程等任务。

Centreon 中检测命令的配置页面在菜单 Configuration → Commands → Checks 中。

如图 11-21 所示, 左侧竖状菜单已经将检测命令按照用途归类, 而列表则显示了系统中所有已经定义的检测命令。

Name	Command Line	Type	Host Uses	Services Uses	Options
<input type="checkbox"/> check_centreon_cpu	\$USER1\$check_centreon_smp_cpu -H \$HOSTADDRESS\$...	Check	0 (0)	0 (1)	1
<input type="checkbox"/> check_centreon_dummy	\$USER1\$check_centreon_dummy -s \$ARG1\$ -o \$ARG2\$...	Check	0 (0)	0 (0)	1
<input type="checkbox"/> check_centreon_load_average	\$USER1\$check_centreon_smp_loadaverage -H \$HOSTAD...	Check	0 (0)	0 (1)	1
<input type="checkbox"/> check_centreon_memory	\$USER1\$check_centreon_smp_memory -H \$HOSTADDRESS...	Check	0 (0)	0 (1)	1
<input type="checkbox"/> check_centreon_ph_connections	\$USER1\$check_centreon_smp_TopConn -H \$HOSTADDRESS...	Check	0 (0)	0 (0)	1
<input type="checkbox"/> check_centreon_nt	\$USER1\$check_nt -H \$HOSTADDRESS\$ -p 12489 -v \$ARG...	Check	0 (0)	0 (0)	1
<input type="checkbox"/> check_centreon_ping	\$USER1\$check_ping -H \$HOSTADDRESS\$ -s \$ARG1\$ -w \$...	Check	2 (0)	0 (2)	1
<input type="checkbox"/> check_centreon_process	\$USER1\$check_centreon_smp_process -H \$HOSTADDRESS...	Check	0 (0)	0 (0)	1
<input type="checkbox"/> check_centreon_remote_storage	\$USER1\$check_centreon_smp_remote_storage -H \$HOST...	Check	0 (0)	0 (11)	1
<input type="checkbox"/> check_centreon_smp					
<input type="checkbox"/> check_centreon_smp_proc_detailed	\$USER1\$check_centreon_smp_process_detailed -H \$H...	Check	0 (0)	0 (0)	1
<input type="checkbox"/> check_centreon_smp_value	\$USER1\$check_centreon_smp_value -H \$HOSTADDRESS\$...	Check	0 (0)	0 (0)	1

图 11-21 检测命令定义

注意：请注意这些检测命令都是以操作系统的 Nagios 用户执行的，因此为了确保检测命令的正确执行，需要确保 Nagios 用户对这些检测命令具备执行权限。

1. 检测命令的校验

在图 11-21 中，单击 Add 链接，可以新增一项检测命令。更方便的新增检测命令的方式是选中一项检测命令，然后在 More actions... 下拉列表中选择“Duplicate(复制)”，像复制出一项服务一样（见 11.2 节），复制出一项检测命令，并在此基础上修改，如图 11-22 所示。

Modify a Command

Check

Command Name * check_centreon_cpu_1

Command Type ☐ Notification ☒ Check ☐ Misc ☐ Discovery

Command Line * \$USER1\$check_centreon_smp_cpu -H \$HOSTADDRESS\$ -v \$HOSTTMPVERSION -c \$HOSTTMPCORENITY4 -c \$ARG1\$ -w \$ARG2\$

Enable shell ☒

Argument Example 18090 \$HOSTADDRESS\$

Argument Descriptions ARG1: critical ARG2: warning

图 11-22 编辑新增的检测命令

如图 11-22 所示，由于新增的 check_centreon_cpu_1 属于 11.2 节中提到的 3 种检测命令中的第 1 种，因此在 Command Type 选项中应该选中 Check 一项，意味着该检测命令是用来检测主机或者服务状态的命令，而非执行通知消息或者其他杂项任务的。

2. Command Line (命令行) 的格式

如图 11-22 所示，在检测命令的 Command Line 文本框内可以定义命令的文本。文本开头通常是存在于 /usr/local/nagios/libexec 目录下的可执行脚本或者可执行程序名称，后跟一系列常规变量和宏（参考 11.4 节）。

绝大多数的检测命令通常以\$USER1\$宏作为开头，该宏在运行时会被检测命令的实际路径，也就是/usr/local/nagios/libexec 所替代。检测命令的格式如下：

\$USER1\$/[可执行检测命令的名称] -H \$HOSTADDRESS\$ [参数]

如图 11-23 所示，为了便于检测命令的书写，Centreon 在命令行文本的右侧提供了 3 项下拉列表供选择：

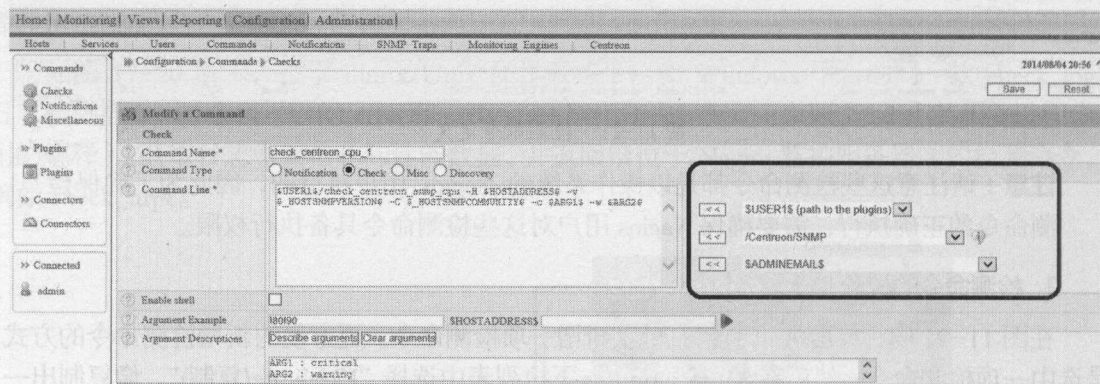


图 11-23 定义检测命令时的下拉列表选项

该 3 项下拉列表依次提供了如下功能：

- (1) 第 1 项提供了一系列环境变量（即 14.4 节中提到的资源宏）；
- (2) 第 2 项提供了检查命令库中所有检测命令的列表，可以从中直接选择检测命令，省去了手工输入的麻烦；
- (3) 第 3 项提供了最常用的一些标准宏，例如主机名、IP 地址、主机组名等等。

举例说明，我们通过在 Command Line 命令文本框内定义了如下检测命令：

```
$USER1$/check_centreon_snmp_cpu -H $HOSTADDRESS$ -C  
$_HOSTSNMPCOMMUNITY$ -v $_HOSTSNMPVERSION$ -W $ARG1$ -c  
$ARG2$
```

该检测命令的任务是通过 SNMP 协议检测被监控主机的 CPU 处理器性能，并根据设置的阈值进行告警。它使用了标准宏\$HOSTADDRESS\$来指定目标主机，使用宏\$_HOSTSNMPCOMMUNITY\$和\$_HOSTSNMPVERSION\$作为 SNMP 命令的参数，使用自定义宏\$ARG1\$和\$ARG2\$分别作为告警阈值和紧急阈值。

3. 参数描述

在定义完检测命令后，接下来的工作就是为该命令的参数添加描述信息，也就是注释信息，以便于后续在调用该检测命令时，能够向用户提示参数的类型和顺序（参见图 11-24）。

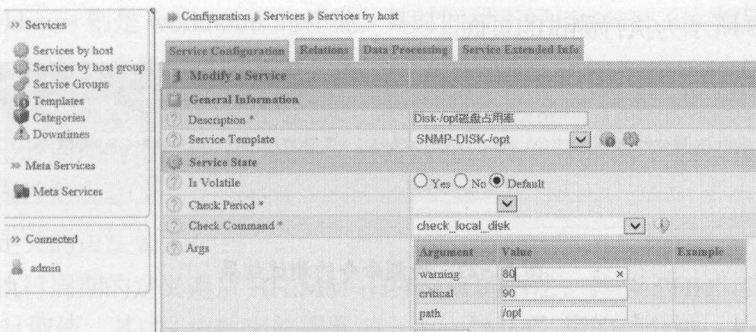


图 11-24 调用检测命令时的参数描述

参数描述信息可以在检测命令定义页面，通过单击 Describe arguments，在弹出的 Argument description 窗口里填写如图 11-25 所示。

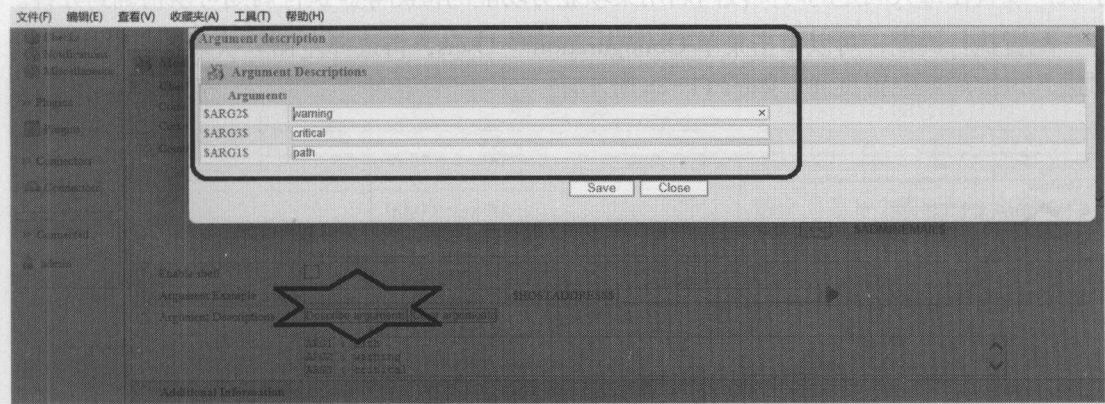


图 11-25 填写参数描述信息

4. 检测命令测试

当检测命令定义完毕之后，可以在该页面的 Argument Example 栏内填入参数，进行检测命令的测试工作，如图 11-26 所示。

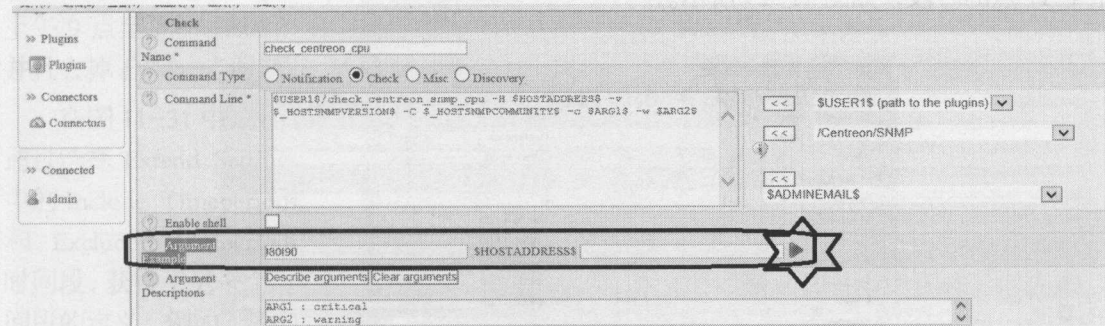


图 11-26 检测命令测试

进行命令测试时，参数列表必须以符号 “!” 作为前缀，且后面的 \$HOSTADDRESS\$ 宏需要填写实际的主机 IP 地址，然后单击箭头图标执行实际的测试工作。

如图 11-27 所示为执行测试的页面。

Plugin Test	
Plugin test	
Command Line	/usr/local/nagios/libexec/check_centreon_snmp_cpu -H 127.0.0.1 -v \$ _HOSTSNMPVERSIONS -C \$ _HOSTSNMPCOMMUNITY\$ -c '80' -w '90'
Output	
Status	CRITICAL

图 11-27 检测命令的测试结果

11.6 执行周期

执行周期(Period)的概念适用于在特定的某一或者某些时间段内禁用或者启用 Centreon 监控系统的某些功能或者特性。例如，在晚上的某个时间段内禁止告警信息发送到管理员的手机；或者在每天固定的时间段内启用某项业务功能的检测等。执行周期可以精确到分钟。

配置执行周期，可以单击菜单 Configuration→ Users，再选择左侧竖状菜单中的 Periods，可进入执行周期配置页面，如图 11-28 所示。

Home Monitoring Views Reporting Configuration Administration	
Hosts Services Users Commands Notifications Nagios Centreon	
Configuration Users Time Periods	
More actions Add	
Name Description Options	
24x7 24 Hours A Day, 7 Days A Week 1	
7_22 7_22 1	
none No Time Is A Good Time 1	
nonworkhours Non-Work Hours 1	
workhours Work hours 1	
More actions Add	

图 11-28 执行周期配置页面

如图 11-28 所示，如果想增加执行周期，可以单击“Add”链接，如果想编辑已有的执行周期，可以点击周期名称链接，进入编辑页面。

Centreon 已经默认提供了一些执行周期定义。例如“24 * 7”，意味着该周期包含了一周内（7 天）的任何时刻（24 小时）。而 workhours 包含的是周一到周五每天的上午 9 点到下午 17 点的时段，如图 11-29 所示。

Configuration Users Time Periods

Basic Settings Extended Settings

Modify a Time Period

General Information

Time Period Name * workhours

Alias * Work hours

Time Range

Sunday

Monday 09:00-17:00

Tuesday 09:00-17:00

Wednesday 09:00-17:00

Thursday 09:00-17:00

Friday 09:00-17:00

Saturday

Time Range exceptions

Exceptions

Days Time Range

List Form

Save Reset

图 11-29 检测周期的 workhours 定义

而 nonworkhours 则是 workhours 的补集，其时间段正好与 workhours 相反。

如图 11-29 所示，在检测周期定义页面的第一个选项卡 Basic Settings 中，提供了周一到周日共 7 个时间槽，可以按照特定格式定义时间段。同时提供了 Time Range exceptions 选项，可以在正常的检测周期内指定某个、或者某些特殊的日志、月份等时刻或者时段，用以正常执行周期内的例外情况。

执行周期定义的语法如下：

- 正常执行周期的定义遵从 HH:MM-HH:MM 的语法，其中“HH”指的是小时，从 01 到 24 取值，其中 1 位数字前需要加上 0。“MM”指的是分钟，从 00 到 59 取值，其中 1 位数字前同样需要加上 0。
- 而例外时间段 Time Range exception 定义时需要遵从特定的语法，具体语法可以参考文档 http://nagios.sourceforge.net/docs/3_0/objectdefinitions.html#timeperiod。

表 11-1 是例外时间段的相关定义。

表 11-1 例外时间段的定义

日期	时间段	描述
2012-08-01	00:00-24:00	2012 年 8 月 1 日全天
friday 4	00:00-06:00	每个月第 4 个周五的 0 点到 6 点
monday -1	12:00-15:00	每个月最后一个周一的中午 12:00 至下午 15:00
july 10 - 20	00:00-24:00	每年的 7 月 10 号至 7 月 20 号的全天
july 10 - august 15	00:00-24:00	每年的 7 月 10 号至 8 月 15 号的全天

例如，图 11-30 定义了名称为 workday 的执行周期，包括周一到周五的上午 9 点至下午 17:00，并且去掉了“十一”假期。

在图 11-31 中，我们通过选择 Extend Settings 中的 Include Timeperiods 和 Exclude Timeperiods 时间段，获得了另一个时间段的定义。例如，我们定义了“下班后的值班时间段”，在 Include Timeperiods 中选择了

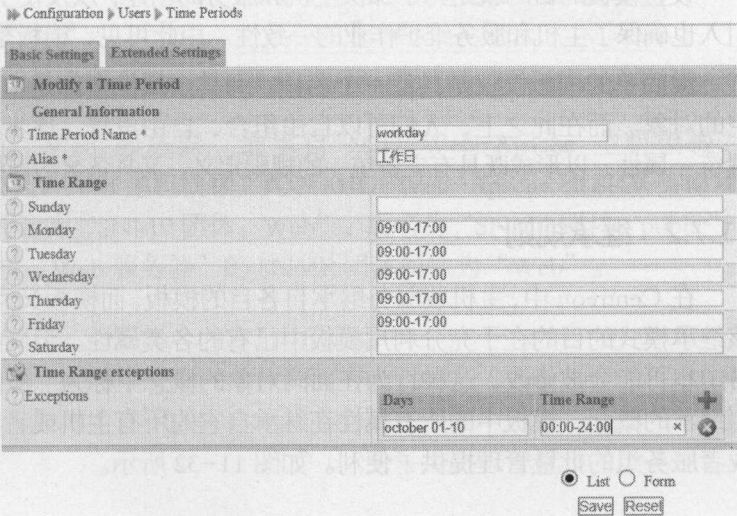


图 11-30 workday 执行周期定义

“24*7”，在 Exclude Timeperiods 中选择了 workhours，如此一来就获得了“下班后的时间段”定义。

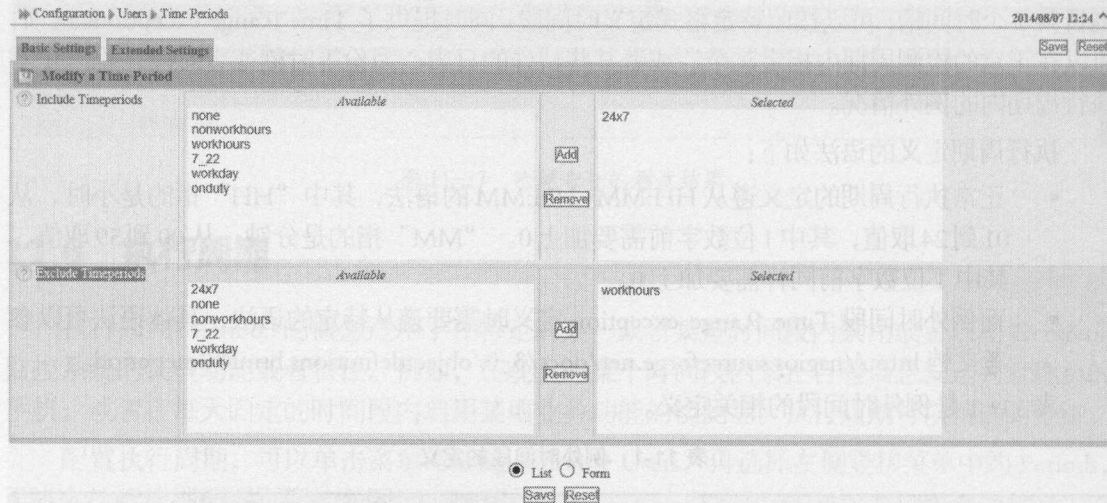


图 11-31 利用 include Timeperiods 和 Exclude Timeperiods 定义时间段

11.7 主机模板和服务模板

11.7.1 模板和继承

模板就是预先设置了通用属性值的主机和服务的载体。通过在模板中设置各种默认属性，使主机和服务的创建变得轻而易举。在对象关系上，主机和服务都是模板的继承者，它们在创建之初就已经具备了各自模板的全部属性，且可以在模板基础上自由修改，从而具备了各自的独立属性。

设置模板的目的就是为了加快主机和服务的创建，以及便于后期的批量维护。模板概念的引入也确保了主机和服务维护作业的一致性。由此可见，模板是 Centreon 的一项伟大的创举。

最简单的模板仅仅是具备一个名称，而其余属性全部是空白的，几乎可以被认为全部为空的对象。而在此之上，人们可以自由组合、自由设置，逐步为后续要定义的主机和服务设置统一属性，以形成既具有完备统一的规则定义、又具备灵活属性的主机群或者服务群。

11.7.2 继承规则

在 Centreon 中，主机或服务继承自各自的模板，而模板又有可能继承自更上一级的模板。该继承模式的目的在于充分利用模板中已有的各类属性。但同样地，模板的属性在主机或服务中也可能被修改，这种行为在面向对象的概念中称为“覆盖或者重载（override）”。按照继承的概念，模板中的所有属性在继承自它的所有主机或者服务中都存在，这就为主机组或者服务组的批量管理提供了便利。如图 11-32 所示。

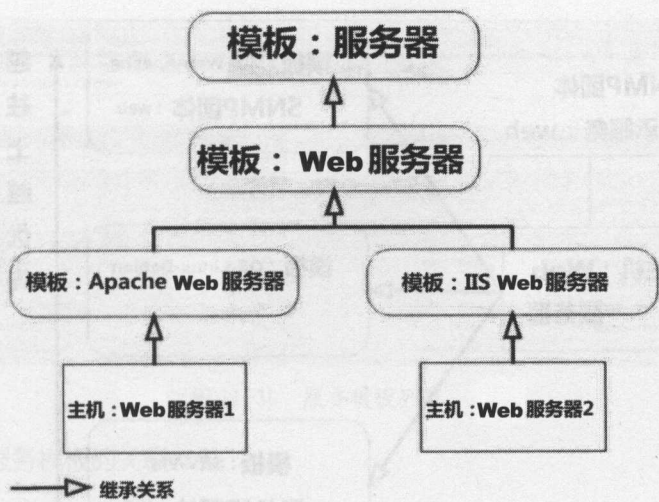


图 11-32 对象和模板之间的继承关系

11.7.3 主机模板

主机模板既可以继承自多个上级模板，也可以什么都不继承，前者意味着该主机模板同时具备多个上级模板中定义的全部属性，而后者意味着该模板自身即为最高级模板，可以有下一级继承者，但不可能再有上级模板。另外，在多模板继承模式中，如果遇到模板间存在冲突的属性，那么具有最高地位的模板的属性优先级会更高。

如图 11-33 所示，某台主机继承了 generic-host 主机模板和 Servers-Linux 模板，且 generic-host 主机模板的属性优先级更高。

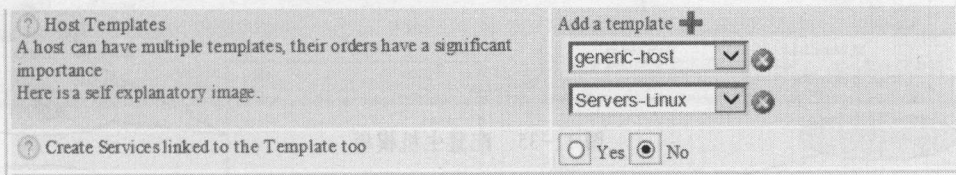


图 11-33 主机模板

如图 11-34 所示，主机“Web 服务器”继承自 MID-Web-Apache 主机模板、OS-Linux_Debian 主机模板和 SRV-VM 主机模板。其中 MID-Web-Apache 和 SRV-VM 都定义了属性“SNMP 团体”，前者为“SNMP 团体：Web”，后者为“SNMP 团体：public”，由于前者优先级更高，因此主机“Web 服务器”的 SNMP 团体属性为“Web”。

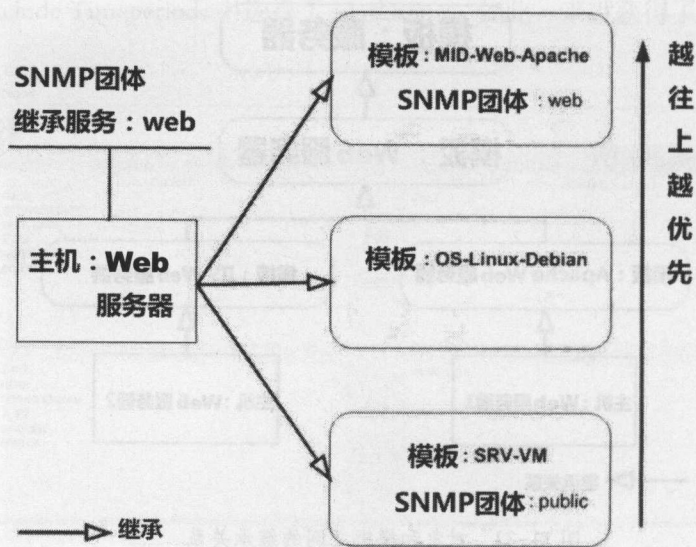


图 11-34 继承优先级示例

要配置主机模板，进入菜单 Configuration → Hosts 如图 11-35 所示。接着单击左侧竖状菜单中的 Templates。

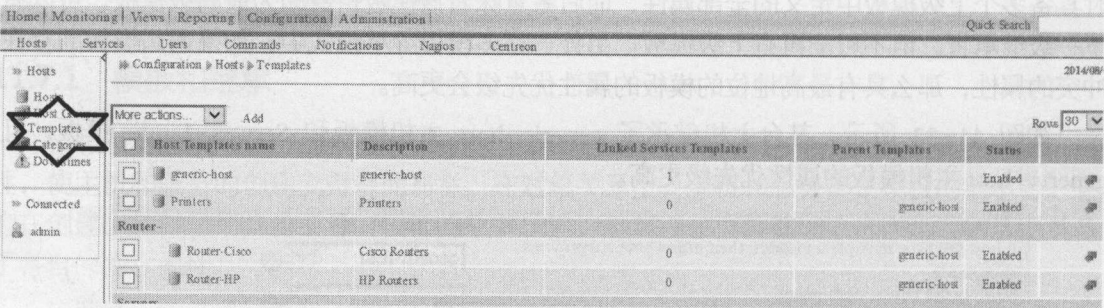


图 11-35 配置主机模板

在图 11-35 中，所有主机模板都以列表的形式存在，且在界面中可以过滤查找、新增模板、复制模板和删除模板。单击任一模板的名称，可进入模板的编辑页面。

1. 服务模板

与主机模板可以继承自多个父模板的模式不同，服务模板仅仅只能继承自一个父模板，无法继承自多个。要配置服务模板，进入菜单 Configuration → Services 如图 11-36 所示，接着单击左侧竖状菜单中的 Templates，可以进入服务模板定义页面，单击模板名，可进入模板的编辑页面。

Home Monitoring Views Reporting Configuration Administration Quick Search						
Hosts	Services	Users	Commands	Notifications	SNMP Traps	Monitoring Engines
Configuration > Services > Templates						
2014/08/08 20:03						
More actions: Add						
Rows 30 Page 1/1						
<input type="checkbox"/>	Service Templates names	Alias	Scheduling	Parent Templates	Status	Options
<input type="checkbox"/>	generic-service	generic-service	5 min / 1 min		Enabled	1
<input type="checkbox"/>	Ping					
<input type="checkbox"/>	Ping-LAN	Ping	5 min / 1 min	-> generic-service	Enabled	1
<input type="checkbox"/>	Ping-WAN	Ping	5 min / 1 min	-> generic-service	Enabled	1
SNMP-DISK- /						
<input type="checkbox"/>	SNMP-DISK- /	Disk- /	5 min / 1 min	-> generic-service	Enabled	1
<input type="checkbox"/>	SNMP-DISK- /home	Disk- /home	5 min / 1 min	-> generic-service	Enabled	1
<input type="checkbox"/>	SNMP-DISK- /opt	Disk- /opt	5 min / 1 min	-> generic-service	Enabled	1
<input type="checkbox"/>	SNMP-DISK- /usr	Disk- /usr	5 min / 1 min	-> generic-service	Enabled	1
<input type="checkbox"/>	SNMP-DISK- /var	Disk- /var	5 min / 1 min	-> generic-service	Enabled	1

图 11-36 服务模板列表

2. 主机模板和服务模板的关联

■ 关联规则

Centreon 支持将主机模板和服务模板关联起来。当基于一个或者多个主机模板创建一个实际主机后，与模板主机相关联的模板服务也同样被实例化（基于面向对象的概念，模板与类（Class）一样，是抽象的概念，类可以被实例化成对象（Object）），形成了一系列与实际主机相关的监控服务器，且名称与服务模板中相应的服务列表名称一致。通过模板以及模板之间的关联，Centreon 实现了主机和服务的快速创建。

如图 11-37 所示：被监控主机 `srv_web` 继承自多个主机模板，且模板主机所关联的所有模板服务均已实例化，并与 `srv_web` 所关联。

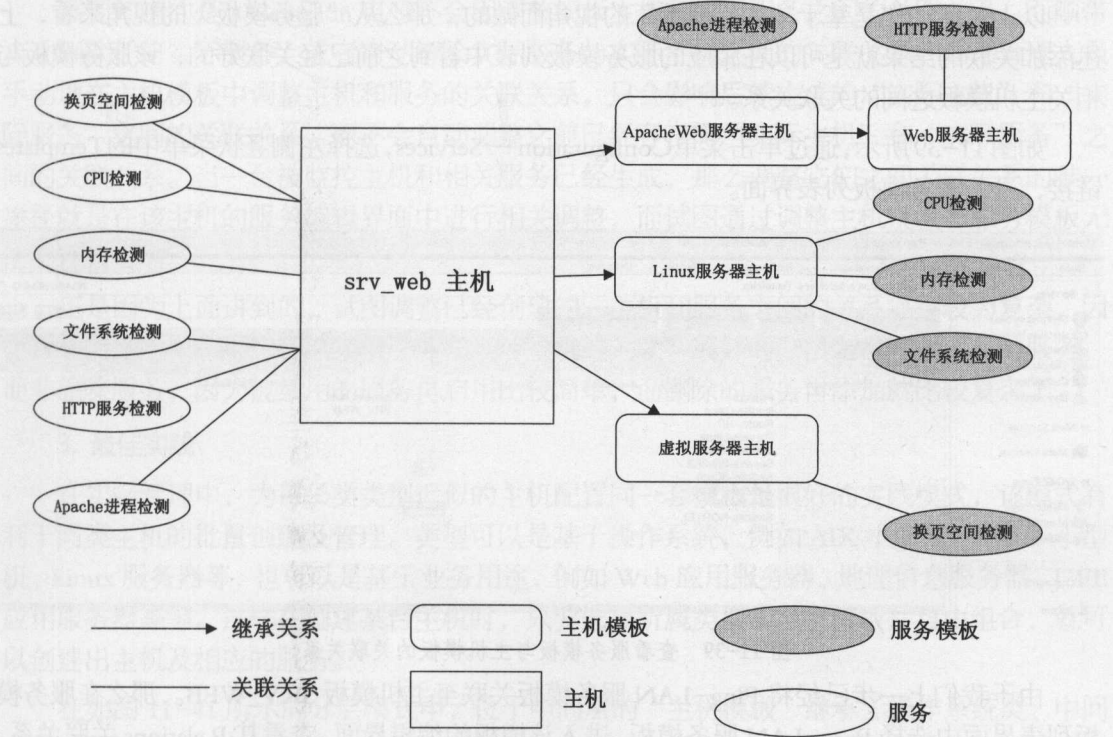


图 11-37 主机模板和服务模板之间的关联关系

■ 配置主机模板和服务模板之间的关联关系

- (1) 单击菜单 Configuration→Hosts，选择左侧竖状菜单中的 Templates 链接，进入主机模板列表界面。
- (2) 在主机模板列表界面单击 Add 链接，进入主机模板增加页面，新增一个名为 SRV_Web 的主机。
- (3) 在 Relations 选项卡，为该新增的主机模板关联合适的模板服务，关联完毕后，可单击 Save 保存，如图 11-38 所示。

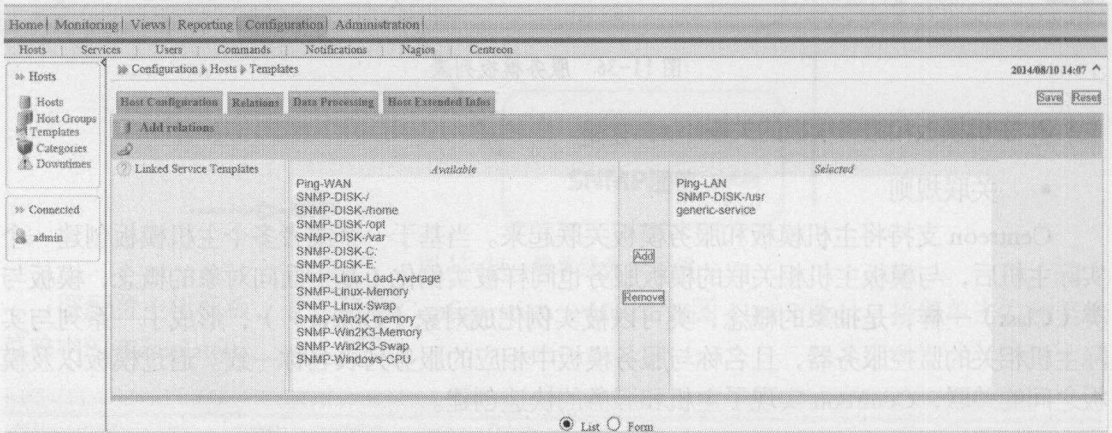


图 11-38 为主机模板关联服务模板

以上关联动作是基于“主机模板”的视角而做的，那么从“服务模板”的视角来看，上述添加关联的结果就是可以在相应的服务模板列表中看到之前已经关联好的，该服务模板与相关主机模板之间的关联关系。

如图 11-39 所示，通过单击菜单 Configuration→Services，选择左侧竖状菜单中的 Templates 链接，进入服务模板列表界面。

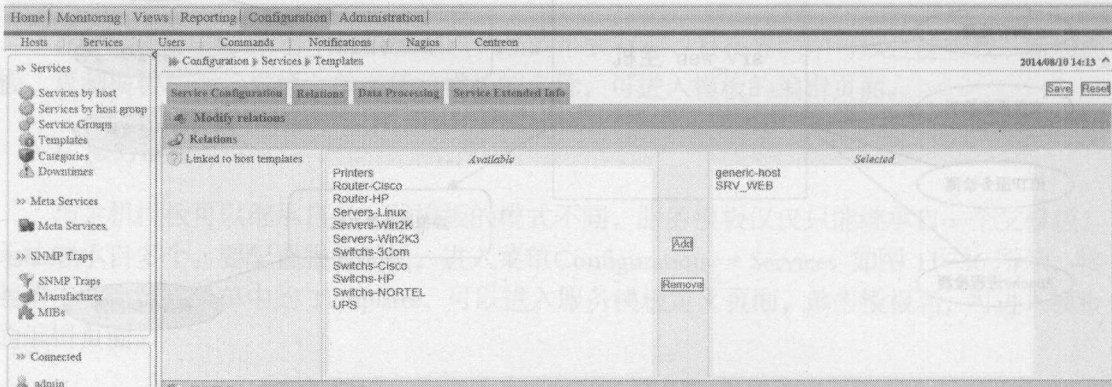


图 11-39 查看服务模板与主机模板的关联关系

由于我们上一步已经将 Ping-LAN 服务模板关联至主机模板 SRV_WEB。那么在服务模板列表界面中选择 Ping-LAN 服务模板，进入该模板的编辑界面，查看其 Relations 关联关系，可以看到该服务模板已经关联到了主机模板 SRV-WEB。

■ 生成服务

在主机编辑界面中，可以选择是否生成相关服务，此选项默认为打开。也就是说，在通过主机模板创建主机时，除了自动生成主机外，Centreon 还可以自动地生成该模板所关联的所有服务。该功能在批量添加被监控主机时非常有用，试想一下，只需一个按钮就可以快速配置一台被监控主机及相关的所有服务，比起手工编辑 Nagios 配置文件并创建主机和服务之间的关联关系来说，生产效率不知提高了多少倍。

为了启用上述功能，需要进入主机编辑页面，选择 Create Services linked to the Template too 项，如图 11-40 所示。

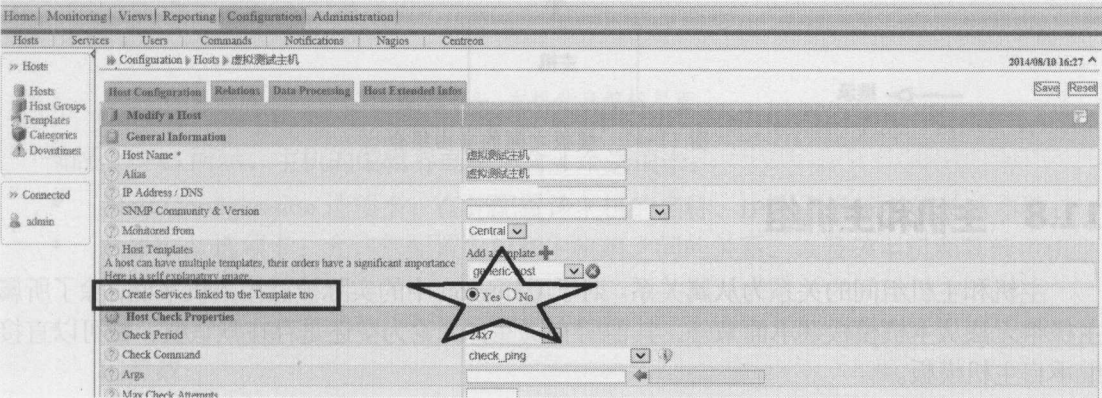


图 11-40 在选择模板的同时创建其服务

需要知道的是，Centreon 自身不会因调整模板主机和模板服务之间的关联关系，而顺带自动删除任何实际服务，或者自动解除实际服务与实际主机之间的关联关系。如果管理人员手动地在主机模板中调整主机和服务的关联关系，只会影响后续生成的“实际主机”和“实际服务”之间的关联关系，而不会自动调整之前已经存在的“实际主机”和“实际服务”之间的关联关系。当一台被监控主机和相关服务已经生成，那么调整它们之间关联关系的唯一途径就是在该主机的服务编辑界面中进行相关调整，而试图通过调整主机模板和服务模板无法实现该目的。

正是因为上面讲到的，试图调整已经创建的、主机和服务之间的关系显得较为复杂，因此日常使用中，如果想解除某项服务和某台主机的关联，最好的办法是禁用该主机的服务，而非删除服务，因为被禁用的服务再启用比较简单，而删除的服务再添加就比较复杂了。

3. 最佳实践

在实际管理中，为每一类类型近似的主机配置同一套模板是很好的实践模式，该模式有利于同类主机的批量创建及管理。类型可以是基于操作系统，例如 AIX 小型机、Solaris 小型机、Linux 服务器等，也可以是基于业务用途，例如 Web 应用服务器、地理信息服务器、J2EE 应用服务器等等。这样在创建某台主机时，只要选择所属类型的主机模板并自由组合，就可以创建出主机及相应的服务。

在如图 11-41 所示的分层模式中，位于中间层的“主机模板”继承了操作系统类、中间件类以及应用类任务的模板，说明该主机模板具备多种功能，而“主机”则直接继承了“主

机模板”，从而将祖先类的多种功能集于一身。那么在创建实际物理主机监控对象的过程中，直接由“主机”模板派生就显得非常方便了。

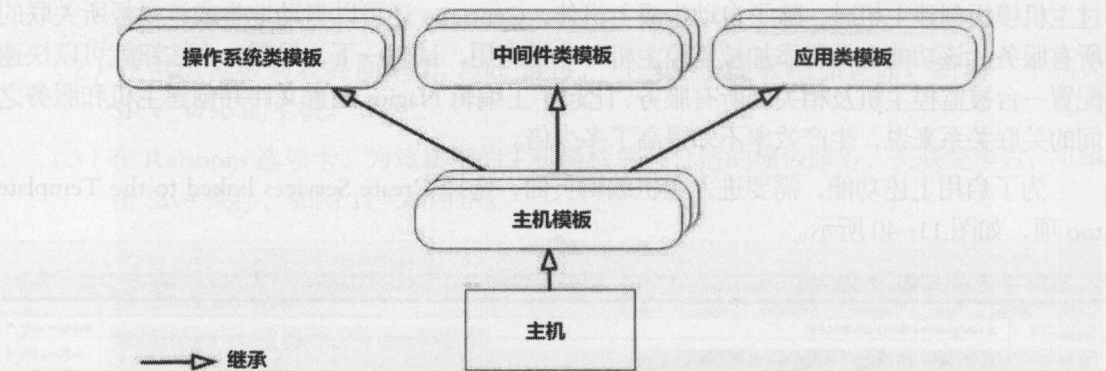


图 11-41 模板之间的自由组合

11.8 主机和主机组

主机和主机组间的关系为从属关系。对于 Centreon 中的实际被监控主机来说，除了所属主机组无法从主机模板继承而来外，其他所有属性，无论为空还是存在默认值，都可以直接继承自主机模板。

在 Centreon 中，对于主机的配置可以通过单击菜单 Configuration→Hosts 进行，如图 11-42 所示显示了 Centreon 监控系统中已经配置的主机列表。

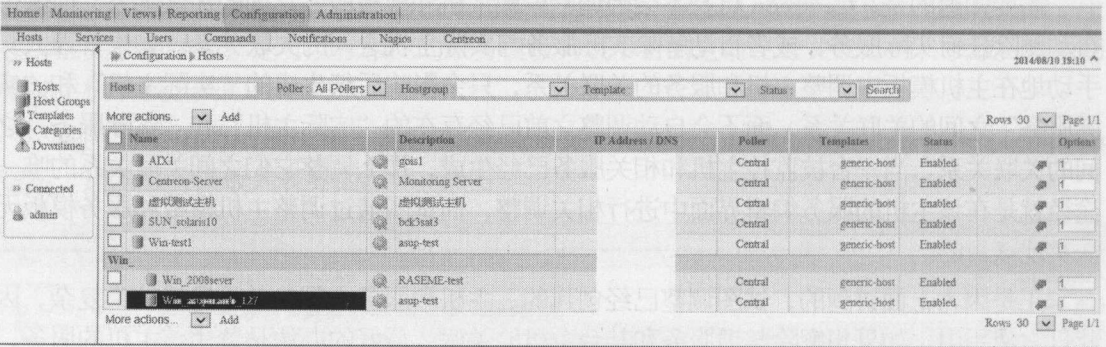


图 11-42 Centreon 中的主机列表

在图 11-42 的 Hosts 文本框中，输入至少 3 位关键字，可支持主机名的全文搜索。在 HostGroup 下拉框中可按照主机组检索主机，还可以单击 Template 下拉列表，按照模板名选择主机，最后，可以按照主机的状态（是否已被禁用）来定位主机。

单击主机列表中的主机名，可进入该主机的信息编辑界面，如图 11-43 所示。

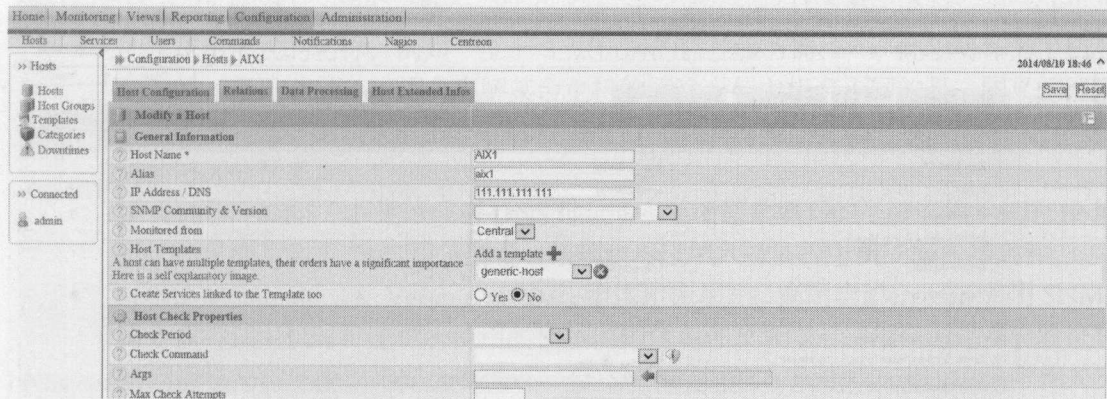


图 11-43 主机信息编辑界面

如图 11-43 所示，主机的编辑界面中包含 4 个选项卡：

- Host Configuration 选项卡：在此配置该主机的名称、IP 以及各项关键检查参数；
- Relations 选项卡：允许您配置主机和主机组之间的关系，以及该主机和其他主机之间的父子关系；
- Data Processing 选项卡：负责配置监控数据、性能数据、历史数据等数据处理相关的参数；
- Host Extend Infos 选项卡：提供了关于该主机的一些补充信息，例如注释、图标、关联文档 URL 等等。

一般来说，配置一台被监控主机，仅仅配置图 11-43 中的前两个选项卡，即 Host Configuration 和 Relations 选项卡就足够了，其余选项保持默认即可。

11.9 主机的配置界面

如图 11-44 所示，是 Centreon 中配置主机相关属性的完整页面：

Home | Monitoring | Views | Reporting | Configuration | Administration

Hosts Services Users Commands Notifications SNMP Traps Monitoring Engines Centreon

Configuration > Hosts > localhostcopy 2014/06/14 21:05

Save Reset

Host Configuration Relations Data Processing Host Extended Info

Modify a Host

General Information

Host Name * localhostcopy

Alias * localhostcopy

IP Address / DNS * 127.0.0.1 Resolve

SNMP Community & Version

Monitored from Central

Host Templates
A host can have multiple templates, their orders have a significant importance
Here is a self explanatory image:
Add a new entry +
Nothing here, use the "Add" button

Create Services linked to the Template too
☐ Yes ☒ No

Host Check Properties

Check Period * 24x7

Check Command check_centreon_ping

Args 131200.20%4400.50%

Max Check Attempts * 5

Normal Check Interval * 60 seconds

Retry Check Interval * 60 seconds

Active Checks Enabled
☐ Yes ☐ No ☒ Default

Passive Checks Enabled
☐ Yes ☐ No ☒ Default

Macros

Custom macros
Add a new entry +
Nothing here, use the "Add" button

Notification

Contact additive inheritance ☐

Linked Contacts *

Available: Guest User
Selected: admin admin
Add Remove

Contact group additive inheritance ☐

Linked Contact Groups *

Available: Supervisors
Selected: Guest
Add Remove

Notification Interval * 1 * 60 seconds

Notification Period * 24x7

Notification Options *
☒ Down ☒ Unreachable ☒ Recovery ☒ Flapping ☒ Devetime Scheduled

First notification delay * 60 seconds

List Form
Save Reset

图 11-44 Centreon 中主机配置选项卡

11.9.1 “通用配置”选项卡

在如上图 11-44 中的主机配置选项卡中，General Information 配置项被用来配置监控主机的常用属性，详述如下。

- Host Name（主机名）：指的是字符串形式的主机名称。该名称不论是在 Centreon 中还是在 Nagios 调度进程中都是独一无二的，不允许重复。在实际命名中，为了便于在 Centreon 中查找并定义某个主机，建议主机名采用“业务名称-服务器名称”的命名方式，例如 Monitor_node1，说明是用于监控用途的，主机名为 node1 的服务器。

- **Alias (同义词)**: 指的是赋予该主机的不同于主机名的、便于记忆的、有解释含义的字符串形式的名称, 例如“位于 3 层机房的监控服务器 1”。此项可作为主机名的互补, 便于 Centreon 用户理解某台主机的用途。与主机名不同的是, 同义词选项并不要求是唯一的, 即多台不同的被监控服务器可以具备完全一致的同义词, 只要您能够区分开来。
- **IP Address / DNS (IP 地址或者 DNS 地址)**: 该项通常填写的是被监控主机的 IP 地址, 如果您用到了 DNS 服务的话, 该项也可以填入被监控主机的服务器名称。
- **SNMP Community & Version (SNMP 团体和版本)**: 指定当 Centreon 采用 SNMP 命令监控该主机时, 主机所使用的 SNMP 团体名称及版本号, 用以通过身份验证和访问授权来实现简单的安全性保证。与 SNMP 相关的术语可以进一步参考相关资料。
- **Monitored from (监控来自于)**: 在采用 Nagios 分布式架构中, 该项用于指定该服务器被哪台 Nagios 调度服务器所管辖。在集中式监控架构中, 该项默认为 Central。
- **Host Templates (主机模板)**: 用于配置被监控主机适用的模板, 可以指定多个模板并设定其优先级关系, 相关知识可以在 11.8 小节中找到。
- **Create Services linked to the Template too (在创建主机时同时创建模板所关联的服务)**: 该项为开关选项, 用于指定在创建被监控主机的同时, 是否创建该主机模板所关联的所有服务。默认为 No。

紧接着的是 Host Check Properties, 即“主机检查属性组”。在该组属性中, 可以设置 Centreon 用以检测主机是否在线的命令, 以及相关检测行为, 详述如下:

- **Check Period (检测周期)**: 该项指定了 Centreon 对于被监控端执行检测行为的周期。多数情况下, 在每个检查周期内, Centreon 都会发起对于主机的检测, 并更新自身对于主机端状态的记录。在周期之外, Centreon 中会保留该主机最近一次的检测状态。
- **Check Command (检测命令)**: 该项定义了 Centreon 用于检测主机是否存活的命令, 一般是 `check_centreon_ping` 命令, 即使用 Ping 命令检测被监控主机是否在线。
- **Args (检测命令参数)**: 在该项中可以指定检测命令的相关参数, 每个参数以 “!” 作为前缀。
- **Max Check Attempts (最大检测次数)**: 定义了 Centreon 在判断某台被监控主机为“宕机”状态之前, 总共发起检测的次数。在实际运行中, Centreon 需要多执行几次检测, 才能最终判断某台主机的实际状态 (Hard 状态/“硬”状态, 翻译成“确定状态”本书采用直接翻译), 该参数即定义了最大的执行次数, 在未达到该最大检测次数之前, 该主机的状态在 Centreon 中被认为是“Soft”状态 (“软”状态, 又称为“待定状态”), 相关定义请参考 11.9 小节。
- **Normal Check Interval (检测间隔)**: 指定了 Centreon 发起对于主机状态的 2 次检测之间的时间间隔, 以秒为单位。
- **Retry Check Interval (重试时间间隔)**: 指的是 Centreon 在初次检测到主机状态发生变化后, 发起再次检测, 直至达到“最大检测次数”, 此期间的检测动作为重试。该选项指定了这些重试动作之间的时间间隔, 注意和“检测间隔”有所不同。

- Active Checks Enabled（启用主动检测）和 Passive Checks Enabled（启用被动检测）：这两项定义了对于被监控对象的检测模式，是采用主动模式还是被动模式。一般来说，检测项都采用主动模式，但对于采用 SNMP trap 方式或者 NSCA 方式发送自身状态到 Nagios 的主机来说，此处应该设置成被动检测模式。

如图 11-45 所示是某台主机的 Host Check Properties 选项组的典型配置：

Host Check Properties	
Check Period *	24x7
Check Command	check_centreon_ping
Args	!3!200,20%!400,50%
Max Check Attempts *	3
Normal Check Interval	5 * 60 seconds
Retry Check Interval	1 * 60 seconds
Active Checks Enabled	<input checked="" type="radio"/> Yes <input type="radio"/> No <input type="radio"/> Default
Passive Checks Enabled	<input type="radio"/> Yes <input checked="" type="radio"/> No <input type="radio"/> Default

图 11-45 配置“主机检测属性组”

如图 11-45 所示，主机检查命令 check_centreon_ping 具有 3 个参数，分别以 3 个“！”作为前缀：

命令参数为“!3!200,20%!400,50%”。其中参数 3 对应了命令检测的次数，参数“200,20%”对应了与网络延迟以及丢包率相关的警告阈值，而参数“400,50%”则定义了网络延迟和丢包率相关的紧急阈值。

而单击 Check Command 项后面的“i”图标，可以显示出实际执行的命令及相关参数项，便于管理人员预览，如图 11-46 所示。

Centreon - IT & Network Monitoring - Internet Explorer	
http://192.168.159.132/centreon/main.php?p=60706&command_id=6&o=w&min=1	
View command definition	
Check command	
Check	check_centreon_ping
Notif	
Command Line	\$USER1\$check_icmp -H \$HOSTADDRESS\$ -n \$ARG1\$ -w \$ARG2\$ -c \$ARG3\$

图 11-46 主机检测命令预览界面

11.9.2 “关系”选项卡

接下来我们来配置主机的“关系”选项卡，在该页面中，有两类关系可以配置，即主机和主机组之间的从属关系，以及主机之间的父—子继承关系。如图 11-47 所示为该选项卡的配置界面。

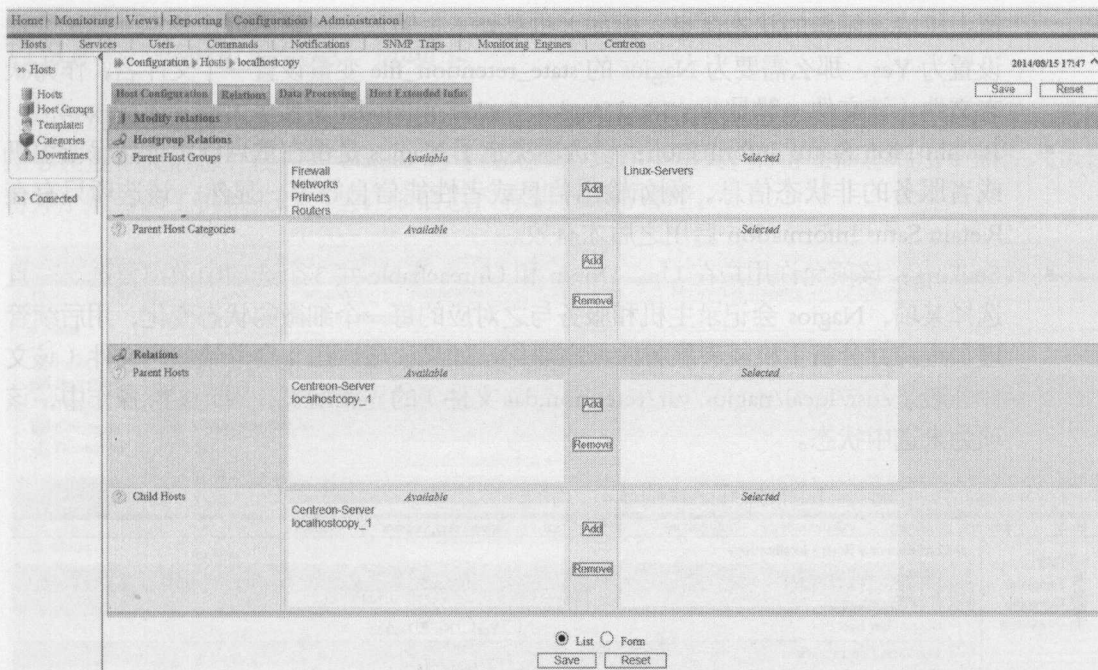


图 11-47 主机的“关系”选项卡

在 Relations 选项卡的 Available 列表中，提供了在 Centreon 中定义的一系列主机组，用户可以选择一个或者多个主机组到 Selected 列表中，将被监控主机指定到所属的主机组。在 Relations 选项卡中，用户可以定义主机之间的父子关系。父子关系特指主机之间的上下级关系，在 Nagios 进行检测时，如果发现父主机发生故障，那么会将子主机标记为“不可达”状态，这样可以避免很多不必要的故障告警信息。

11.9.3 “数据处理”选项卡

如图 11-48 所示为该选项卡的配置界面。

- **Obsess Over Hosts Option:** 该选项决定了 Nagios 是否对于主机的检测结果进行“纠缠”，并且执行预先定义的 `obsessive compulsive host processor command` 命令（该命令允许 Nagios 在执行每一次主机检测动作后执行一个预先定义的命令，在 Nagios 分布式监控模式下经常会用到）。该选项在 Nagios 分布式架构下有用，如果采用集中式监控，请将该项设置为 No。
- **Check Freshness:** 该选项定义了被动监控模式下，Nagios 是否需要定期执行检测命令以确保主机的状态是最新的。而 Freshness Threshold 则定义了 Nagios 的检测频率。
- **Flapping:** 定义了是否对该主机启用抖动检测。一旦启用抖动检测，当 Nagios 探测到主机或者服务的状态发生频繁变化后，为避免发出过多的告警通知消息，会临时性地停止发送告警消息，直到主机或者服务的状态恢复稳定。正因为如此，抖动检测是一个完全凭经验度量的参数，Low Flap Threshold 和 High Flap Threshold 定义了抖动检测的阈值，单位是百分比。
- **Retain Satus Information:** 该选项决定了 Nagios 在发生重启动作之间，是否依然保

留主机或者服务的状态信息，例如主机是处于告警状态还是不可达状态。如果该项设置为 Yes，那么需要为 Nagios 的 state_retention_file 变量设置一个文件名，作为状态文件（该文件一般是 /usr/local/nagios/var/retention.dat 文件）。

- **Retain Non Status Information:** 该选项决定了 Nagios 是否在重启以后依然保留主机或者服务的非状态信息，例如输出信息或者性能信息等等。显然，该选项只有在 Retain Satus Information 启用之后才有效。
- **Stalking:** 该项允许用户在 Up、Down 和 Unreachable 共 3 个选项中做出复选。一旦选择某项，Nagios 会记录主机和服务与之对应的每一个细微的状态变化，用后续管理员跟踪并分析主机或者服务的状态变化。如果该项启用，会导致状态文件（该文件一般是 /usr/local/nagios/var/retention.dat 文件）的显著增长，因此实际操作中，该项为未选中状态。

图 11-48 数据处理选项卡

注意：需要注意的是，在以上选项中，Stalking 选项允许 Nagios 调度进程记录主机的细微变化，因此在查找主机问题的时候会经常用到，但由于日志文件的增大，会显著降低在 Centreon 的 Web 界面查看主机日志的速度。而其他选项在 Nagios 中为全局参数，一般不必为某个主机特别设置，保持默认即可。而如果确实想修改这些参数中的某些项目，只需要在主机模板中修改，然后适用于某主机即可，而不必在 Nagios 全局配置文件中修改，这样可以避免对整个 Centreon 和 Nagios 监控平台产生全局性的影响。

11.9.4 “主机扩展信息” 选项卡

该选项卡主要用来设置主机的图形信息、备注信息等附加信息，如图 11-49 所示。其中图形设置相关信息会在一些 Nagios 的可视化图形组件中用到，而备注信息则用于监控人员查看并了解与该主机相关的一些详细信息，有助于日常管理。

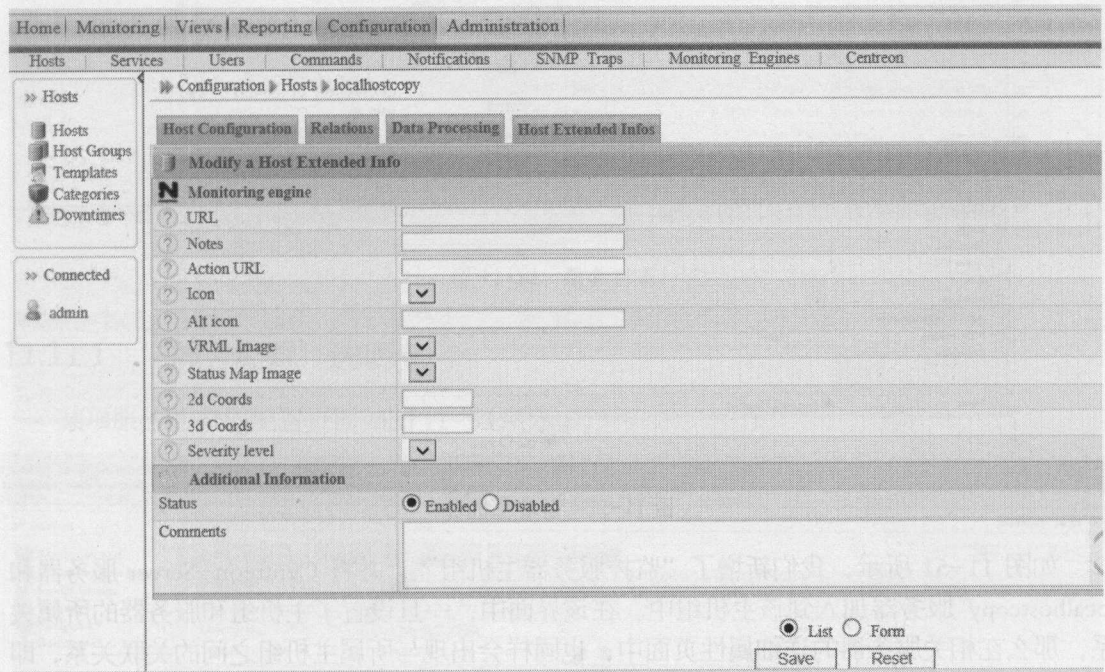


图 11-49 主机扩展选项卡

11.10 主机组

可以单击菜单 Configuration → Hosts，接着单击左侧竖状菜单的 Host Groups 项，进入主机组列表界面，如图 11-50 所示。

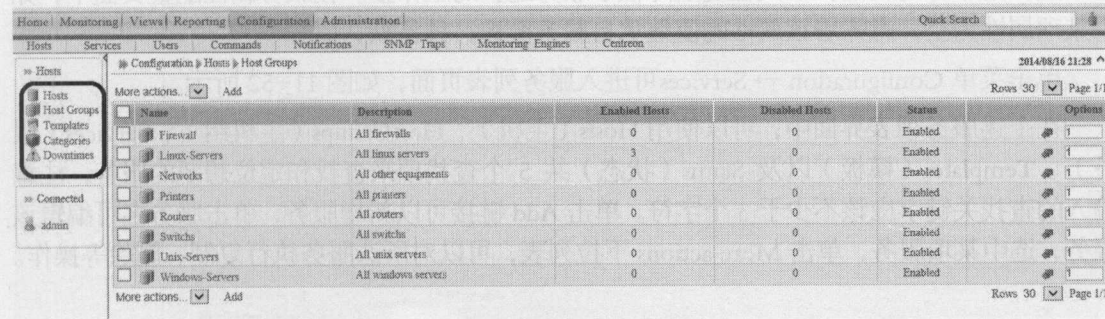


图 11-50 查看主机组列表

单击上图中的 Add 按钮，可进入主机组添加页面，如图 11-51 所示。

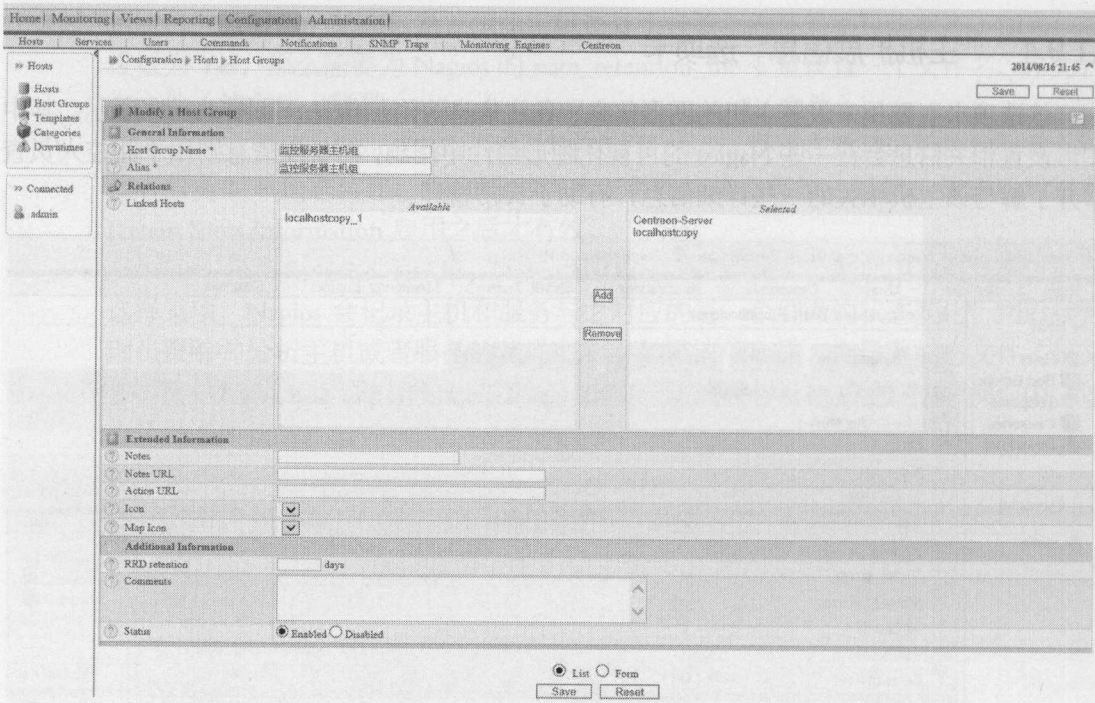


图 11-51 新增主机组

如图 11-51 所示，我们新增了“监控服务器主机组”，并将 Centreon-Server 服务器和 localhostcopy 服务器加入到该主机组中。在该界面中，一旦设置了主机组和服务器的所属关系，那么在相关服务器的详细属性页面中，也同样会出现与所属主机组之间的关联关系，即该关系在主机组页面及其相关主机的详细属性页面是完全一致的。

11.11 服务

一般来说，除了服务与服务组之间的关联关系外，服务的所有属性都可以来自于服务模板。换句话说，如果定义好服务模板，后续就可以使用这些模板派生出大量服务，且不用逐个配置派生服务的属性，很大程度上降低了服务配置的工作量。在服务属性配置页面中，如果某项属性为空，意味着该属性的值继承自服务模板，具备默认值。

单击菜单 Configuration → Services 可进入服务列表页面，如图 11-52 所示。

在上述服务列表界面中，可以使用 Hosts（主机）、HostGroups（主机组）、Services（服务）、Templates（模板）以及 Status（状态）共 5 个查找项来查找和定位想要的服务，注意输入的查找关键字应该不少于 3 个字符。单击 Add 链接可以新增服务，单击服务名可编辑该服务，选中某项服务，单击 More actions 下拉列表，可以对所选服务执行复制、删除等操作。

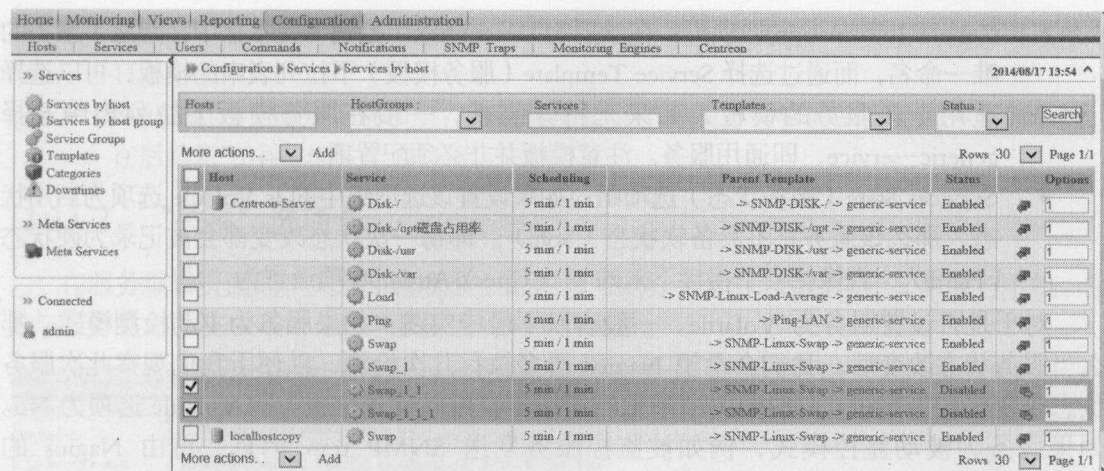


图 11-52 服务列表

11.11.1 “服务配置” 选项卡

某项服务的典型配置页面如图 11-53 所示。

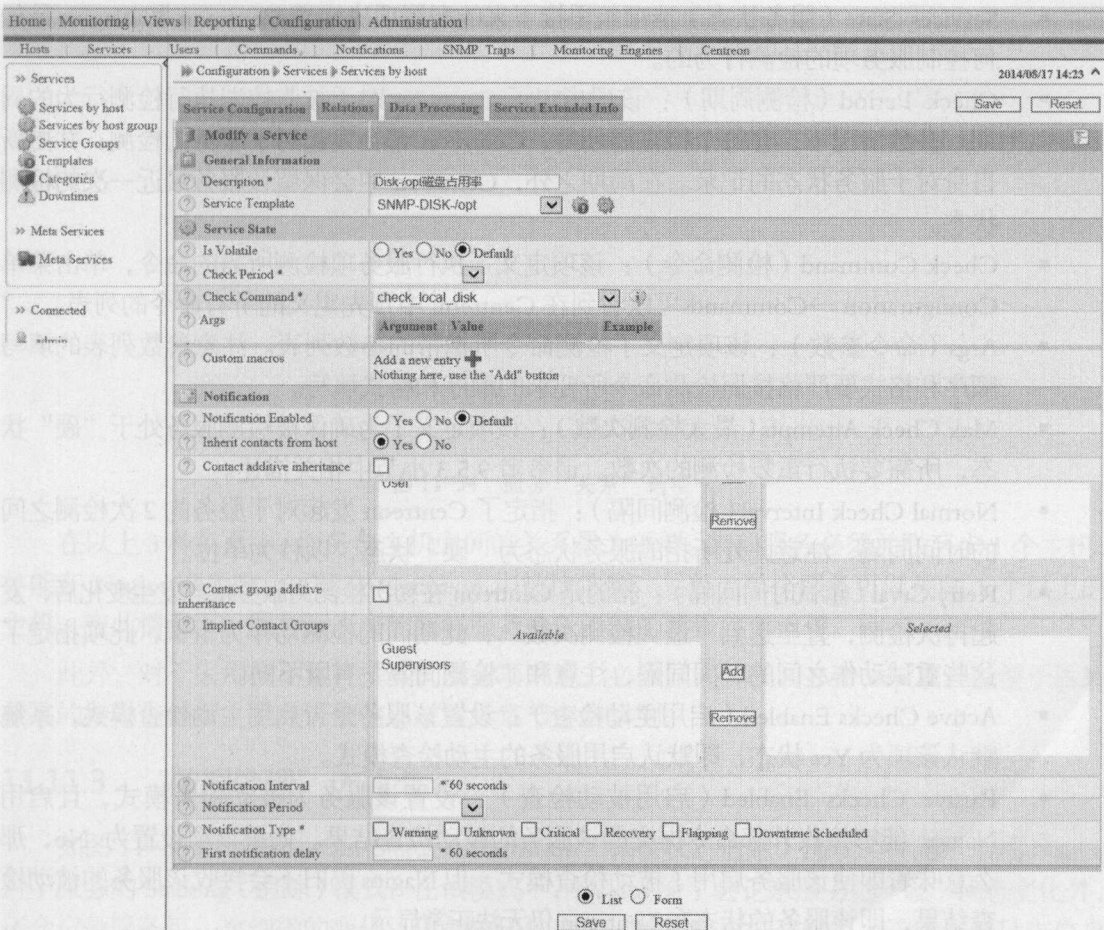


图 11-53 服务编辑页面

- **General Information 选项组**：在该项的 Description (描述) 文本框中，定义了服务的唯一命名，而通过选择 Service Template (服务模板) 下拉列表中的模板，可以选择适用于该服务的模板，若无特殊需求，一般在服务模板下拉列表中选择 generic-service，即通用服务。注意模板并非必须配置项。
- **Services State (服务状态) 选项组**：如果设置该选项组中的 Is Volatile 选项为启用状态，那么该服务将不具备软状态 (Soft)。即每一次状态改变都会被记录为硬状态 (Hard)，就像相当于将接下来的 Max Check Attempts 项设置为 “1”。

对于是否设置服务为 Volatile，一般有以下最佳实践：如果服务为主动检测模式，那么当服务状态改变后，我们会希望 Nagios 再多执行几次检测，以便于我们观察此次服务状态改变是否为误报，同时也为了降低告警频率，可以设置该服务的 Volatile 选项为 No。如果服务为被动监控模式，例如被监控服务发送 SNMP trap 消息，再由 Nagios 的 submit_check_result 脚本提交给 Nagios 主控端的监控方式，为了我们能够在服务每一次产生状态变化，例如从正常状态变成告警状态、或者从紧急状态下恢复正常时，都可以接收到通知消息，需要将该服务的 Volatile 选项设置为 Yes。一般来说，可以在服务模板中针对服务类型设置 Volatile 选项，然后根据服务采用的主动模式还是被动模式再选择合适的模板即可。

- **Services State (服务状态) 选项组**里接下来的配置项决定着 Centreon 和 Nagios 是如何控制服务项的检测行为的。
- **Check Period (检测周期)**：该项指定了 Centreon 对于被监控端执行检测行为的周期。多数情况下，在每个检查周期内，Centreon 都会发起对于服务的检测，并更新自身对于服务状态的记录。在周期之外，Centreon 中会保留该服务最近一次的检测状态。
- **Check Command (检测命令)**：该项定义了执行服务项检测所需的命令，单击菜单 Configuration → Commands 可以看到在 Centreon 中预先定义的所有命令的列表。
- **Args (命令参数)**：该项定义了检测命令所携带的参数列表。注意参数列表的填写顺序和格式要严格按照检测命令所要求的顺序和格式填写。
- **Max Check Attempts (最大检测次数)**：该项定义为确保被检测服务处于 “硬” 状态，所需要执行重复检测的次数。请参考 9.5.3 小节中相关描述。
- **Normal Check Interval (检测间隔)**：指定了 Centreon 发起对于服务的 2 次检测之间的时间间隔，注意此处所指的服务状态为 “硬” 状态，以秒为单位。
- **Retry Crval (重试时间间隔)**：指的是 Centreon 在初次检测到服务状态发生变化后，发起再次检测，直至达到 “最大检测次数”，此期间的检测动作为重试，此项指定了这些重试动作之间的时间间隔，注意和 “检测间隔” 有所不同。
- **Active Checks Enabled (启用主动检查)**：设置该服务是否启用主动检查模式，系统默认该项为 Yes 状态，即默认启用服务的主动检查模式。
- **Passive Checks Enabled (启用被动检查)**：设置该服务为被动检查模式，且启用 Nagios 能够接收外部命令提交的该服务的被动检查结果。如果该项设置为 No，那么意味着即使该服务启用了被动检查模式，但 Nagios 仍旧不会接收该服务的被动检查结果，即该服务的状态在 Centreon 仍无法正常显示。

- Macros（宏）：在此选项组里允许用户定义一些自定义宏，有关宏的概念可以参考 11.4 小节。
- Notification（通知）选项组：该选项组定义了与服务告警通知有关的一系列参数，在后续章节中有相关详细讨论。

11.11.2 “关系” 选项卡

在服务编辑页面的“关系”选项卡（如图 11-54 所示）中，您可以定义服务与主机、服务与服务组，以及 SNMP trap 与服务之间的关联关系。

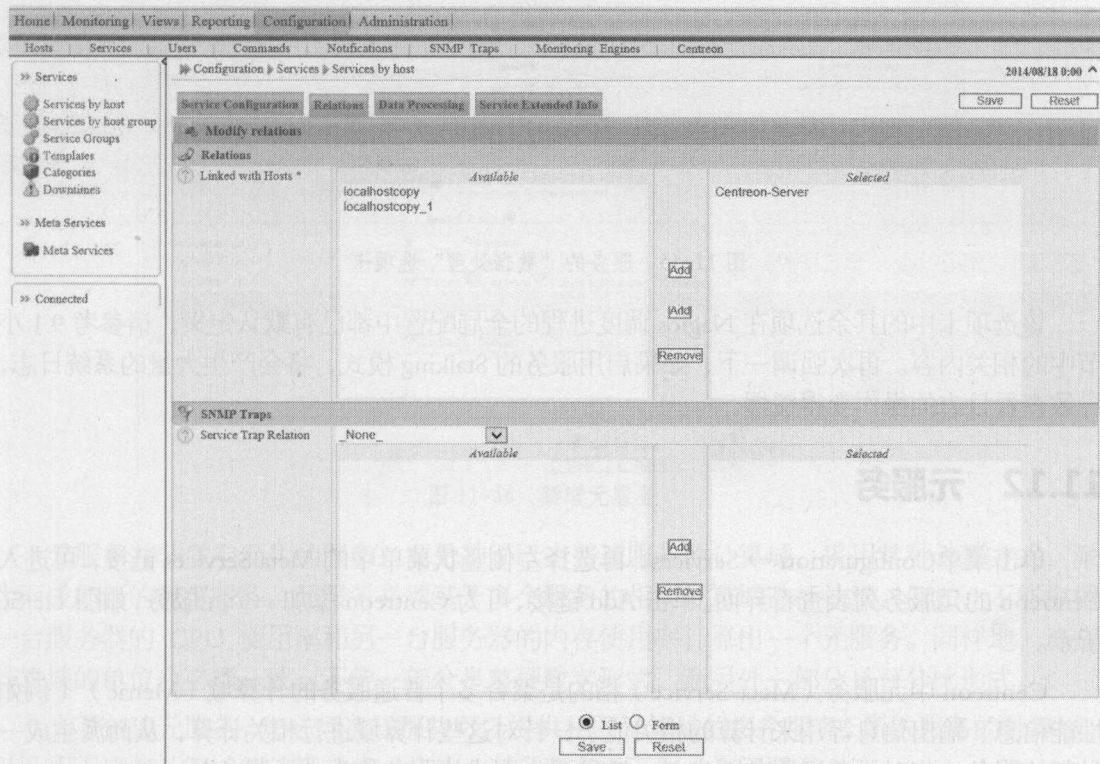


图 11-54 服务“关系”选项卡

在以上 3 种关系中，服务与主机之间的关系最为重要。某项服务必须关联至少 1 个主机，才能在 Nagios 中生效，才能在 Centreon 中被检索到。另外，单个服务可以关联 1 个以上的主机，在此情况下，服务列表中的同样服务会以不同的背景底色出现。

此外，对于采用被动监控模式的服务项来说，必须显式关联 SNMP trap 项，以便于后续状态的接收与处理。相关详细信息在后续章节中会讲到。

11.11.3 “数据处理” 选项卡

如图 11-55 所示的 Retain Status Information（保留状态信息）选项的作用是启用 Nagios 对于服务的 Stalking（追踪）模式。在该模式下，Nagios 除了会记录服务的“硬”状态变化外，还会记录服务每一次细微的变化，例如输出的变化、性能数据的变化等。这些详细日志信息

的记录有助于服务问题的排查和解决。

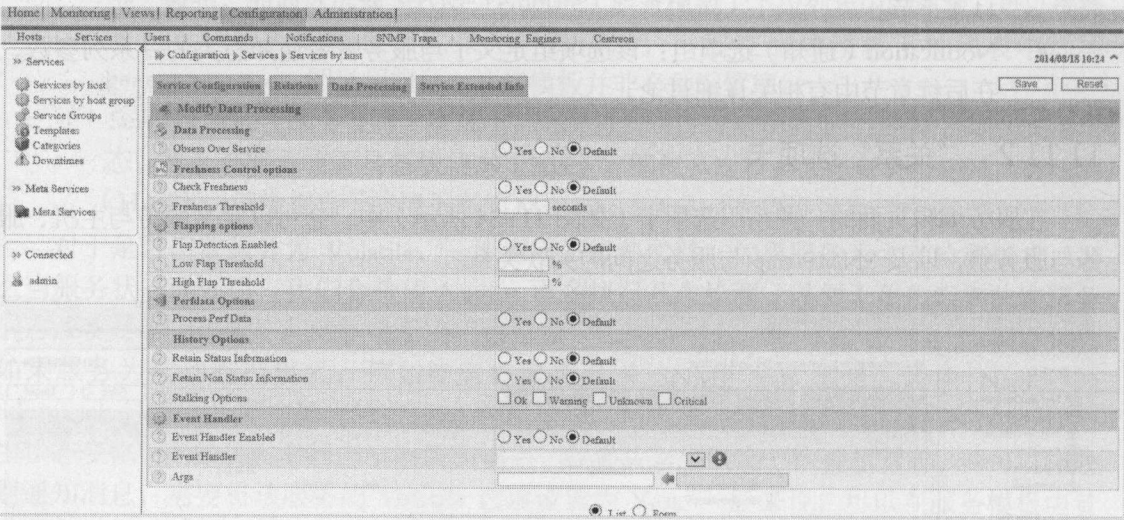


图 11-55 服务的“数据处理”选项卡

该选项卡中的其余选项在 Nagios 调度进程的全局配置中都已默认定义，请参考 9.1 小节中的相关内容。再次强调一下，如果启用服务的 Stalking 模式，将会产生大量的系统日志，导致查看日志的操作变得缓慢。

11.12 元服务

单击菜单 Configuration → Services，再选择左侧竖状菜单中的 Meta Services 链接，可进入 Centreon 的元服务列表查看界面。单击 Add 链接，可为 Centreon 添加一个元服务，如图 11-56 所示。

Centreon 中元服务（Meta Service）指的是聚合多个普通服务的计算域（Metric）（例如性能信息、输出信息等用来计算的相关值），针对这些计算域进行相关计算，从而派生成一种新的服务（相对于普通服务而言）。这种派生而成的服务称为“元服务”。

以上概念不是很好理解，接下来我们将借助于两个例子，从系统管理人员的角度了解元服务的用途。

- 某三层架构的应用系统中，具备多台 JBoss 应用服务器，以负载均衡的方式对外提供服务，如果其中一台 JBoss 服务器 CPU 负载较高，并不影响其余 JBoss 服务器的运行，也能够正常提供对外服务。那么我们就可以根据这些 JBoss 服务器的 CPU 负载值派生出一个虚拟的 JBOSS 应用服务器系统的 CPU 负载值，作为元服务使用。这种方式比为每台 JBoss 服务器单独配置 CPU 负载监控要更有效率，且管理人员收到的告警信息会更少。
- 还是同样的例子，在多台 JBoss 服务器中，每台服务器都对外提供了多个访问会话。那么将每台服务器上的会话数量聚合成一个会话总数元服务，进行相应的检测和配置告警信息，就比为每台服务器配置会话数量检测和配置告警要更有效率，也更节省监控日志。

Home | Monitoring | Views | Reporting | Configuration | Administration |

Hosts | Services | Users | Commands | Notifications | SNMP Traps | Monitoring Engines | Centreon

Configuration > Services > Meta Services

2014/08/19 11:14

Save Reset

Add a Meta Service

General Information

Meta Service Name * Ping元服务

Output format string (printf-style) Ping命令的输出格式为 %s 毫秒

Waiting Level 5

Critical Level 10

Calculation Type * Sum

Data Source Type GAUGE

Selection Mode * ☒ Service List ☐ SQL matching

SQL LIKE-clause expression

Metric

Meta Service State

Check Period * 24x7

Max Check Attempts * 3

Normal Check Interval * 5 * 60 seconds

Retry Check Interval * 5 * 60 seconds

Notification

Notification Enabled * ☐ Yes ☒ No ☐ Default

Linked Contact Groups *

Available: Guest

Supervisors: Selected

Buttons: Add, Remove

Notification Interval * 5 * 60 seconds

Notification Period * 24x7

Notification Type * ☒ Warning ☒ Unknown ☒ Critical ☐ Recovery ☒ Flapping

Additional Information

Graph Template Latency

Status ☒ Enabled ☐ Disabled

Comments

List Form

Save Reset

图 11-56 新增元服务

元服务并非凭空而生的服务，而是基于已有普通服务的计算域，采用某种计算方式，派生而成的新的虚拟服务。元服务必须基于多个服务的相同计算域进行计算，例如，不能根据一台服务器的 CPU 使用率和另一台服务器的内存使用率计算出一个元服务。同样地，这些计算域的单位也必须一致，不能一部分是整型数字形式，而另外一部分是百分比形式。

为了创建元服务，您必须选择那些可以提供相同计算域的多个服务，所选服务的计算域，以及用于这些计算域的计算方式。此外，还需要考虑下面 3 个问题：

(1) 如何选择用于计算元服务的这些普通服务？

(2) 什么是服务的计算域？

(3) 计算公式是什么？

第二个问题比较简单，所谓计算域（Metric）即用来提交计算的，所选服务的某项性能数据。第三个问题即用于这些计算的数学表达式，可以是求和、可以是求平均数等等。而对于第一个问题，则提供了如下两种答案：

答案一：选择具备同样服务名或者类似服务名的服务。在 Centreon 提供的元服务配置界面中，对应于选项 Selection Mode（选择模式）中的 SQL matching（SQL 匹配）项。使用此类服务选择方式，所选服务需要具备同样名称或者具备同样关键字，Centreon 使用服务关键字在后台 MySQL 数据库中查找所匹配的服务，将所匹配的服务选择出来并进行后续计算。例如，所有具备 SessionNumber（会话数量）关键字的服务将通过匹配 %SessionNumber% 的方

式被选取，并据以计算出总的会话数量。

答案二：采用“单独定义的服务列表”方式，手工选择特定的服务，共同组成用于计算的服务列表。在 Centreon 提供的元服务配置界面中，对应于选项 Selection Mode（选择模式）中的 Service List（服务列表）项。这种服务列表方式适用于提供同种类型服务的主机群组里，例如 JBoss 应用服务器组，在上述服务器组中，服务名可以不具备同样的关键字，只要是提供同样的服务，输出同样的计算域即可。

接下来我们进入 Centreon 提供的元服务相关配置页面，详细介绍元服务的有关特性，及配置元服务的相关方法。

如图 11-56 所示，元服务属性的配置页面与普通服务的属性配置页面类似，但是也有一些细微的不同。详细的配置项如下所述：

- Meta Service Name（元服务名）：规定了系统中唯一的元服务名称，为字符串格式。
- Output format string (printf-style)（输出字符串格式（打印格式））：规定了服务输出信息的打印格式，适用于标准的打印通配符。例如，变量“d%”可以被与服务相关的度量值所代替，如下例子：
“到网站 www.monitor.com 的 Ping 值延迟为 d%毫秒”
其中的“d%”可以被 http 检测插件返回的真实的 Ping 值延迟所替代。
- Warning Level（警告阈值）和 Critical Level（紧急阈值）：这两个绝对值定义了元服务的警告阈值和紧急阈值，一般为可以度量的数字或者百分比格式。
- Calculation Type（计算类型）：提供了可供选择的服务计算域的计算方式，有 Average（平均值）、Sum（求和）、Min（最小值）和 Max（最大值）共 4 种计算类型可供选择。
- Selection Mode（选择模式）：设定了用以进行后续计算的服务的提取方式。在此提供了两种方式：第一种是手工选择特定的服务，从而形成服务列表，第二种是基于 SQL 匹配的方式动态提取服务列表。第二种方式需要提供一个 SQL 表达式供系统从后台 Mysql 数据库中提取服务列表所用。
- SQL LIKE-clause expression（SQL Like 子句）：此项只有当选择了 Selection Mode（选择模式）项中的 SQL matching 才有效。在该项中，可填入适用于 Mysql 数据库的 SQL 表达式，格式一般为“%”形式的 Like 子句。例如，Traffic%或者 CPU%。
- Metric（计算域）：提供了用于度量元服务状态的计算域列表，例如文件系统（/opt、/usr 等）、系统负载（load1、load5、load15 等），以及其他可供度量的系统运行参数。

以下分别是采用 SQL matching 方式和采用 Service List 方式组装元服务的例子：

1. SQL matching 方式

为了计算出多台 JBoss 的总会话数量，需要动态提取单台 JBoss 服务器中，具备 SessionNumber 关键字的服务，这些服务都具备名为 SessionNumberOutput 的输出。那么在新增元服务的页面中，相关项应如下配置：

- Selection Mode : SQL matching。
- SQL LIKE-clause expression : SessionNumber。

▪ Metric : SessionNumberOutput。

2. 手动选择服务列表方式

手动选择服务列表必须在元服务已经添加完毕之后才能进行，以下是配置步骤：

步骤一：新增 MetaService 元服务，此时可以在元服务列表中看到刚刚新增的元服务，如图 11-57 所示。

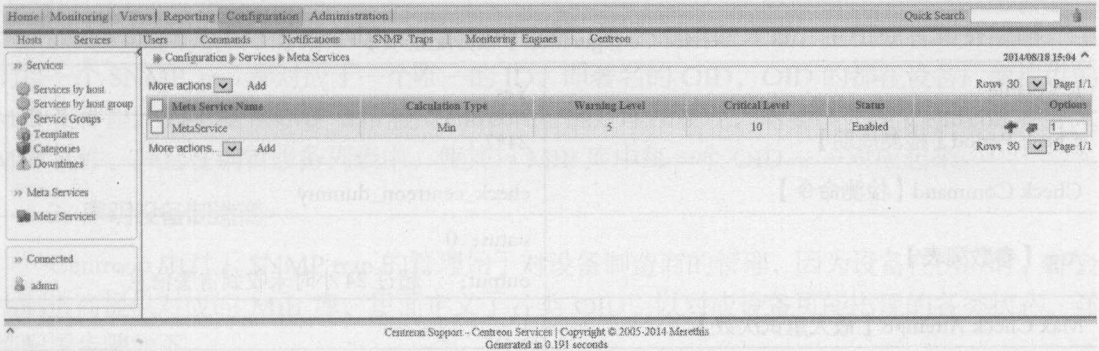


图 11-57 新增元服务

步骤二：单击该新增 MetaService 元服务后面的“+”图标，进入服务选择界面，单击 Add 链接，进入手工选择服务页面。如图 1-58 所示，该界面可以允许用户手动选择可供计算的服务，形成服务列表，并用以计算元服务。

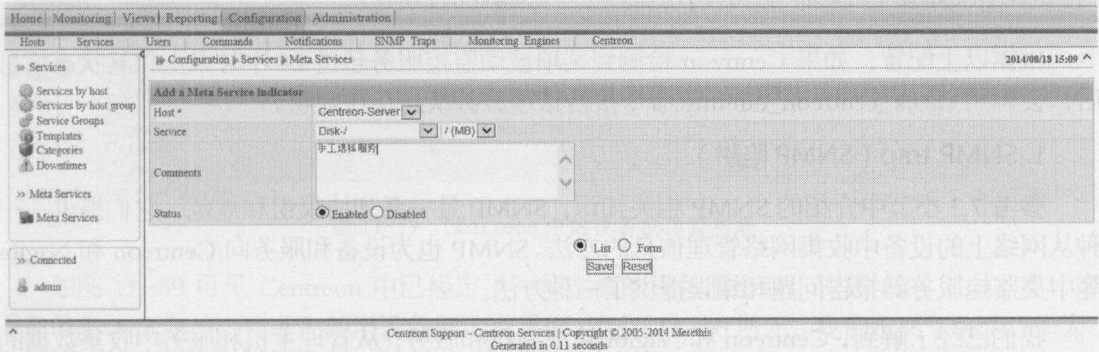


图 11-58 手工选择服务列表

11.13 被动监控模式和 SNMP trap (SNMP 陷阱)

在 Centreon 中，采用被动监控模式的服务具备如下特点：

如 11.11 小节所述，采用被动监控模式的服务在 Centreon 中被配置成 Volatile 的，意味着被动监控服务的每一个状态变化都会被 Centreon 记录为“硬”状态，并会发出告警通知信息。

Centreon 不需要为被动监控模式下的服务配置检测命令（检测探针），但仍旧需要为这些服务指定一个特殊命令，用以检测服务的状态是否时刻在刷新。

一般地，Centreon 中的被动监控服务一般都是采用 SNMP trap 机制，即 Centreon 接收网络设备发送的 SNMP trap 信息，交由后台进程处理后根据预先的配置做告警处理。

采用被动监控模式的服务一般使用如下的属性配置项，如表 11-2 所示。

表 11-2 被动监控模式服务属性表格

服务属性	推 荐 值
Service Configuration【服务配置项】	
Is Volatile	Yes
Check Period【检测周期】	24*7
Check Command【检测命令】	check_centreon_dummy
Args【参数列表】	status: 0 output: “超过 24 小时未收到告警信息”
Max Check Attempts【最大重试次数】	1
Active Checks Enabled【启用主动检测】	No
Passive Checks Enabled【启用被动检测】	Yes
Data Processing【数据处理选项卡】	
Check Freshness【是否检查最新结果】	Yes
Freshness Threshold【结果新鲜度阈值】	86400（24 小时）

根据以上配置，如果 Centreon 检测到某项被动监控服务超过 24 小时未上报其状态信息后，会调用 check_centreon_dummy 命令报告该项服务状态存在问题。

1. SNMP trap（SNMP 陷阱）

参考 7.3 小节中介绍的 SNMP 相关知识，SNMP 是一系列协议组和规范，它们提供了一种从网络上的设备中收集网络管理信息的方法。SNMP 也为设备和服务向 Centreon 和 Nagios 等中央监控服务器报告问题和错误提供了一种方法。

我们已经了解到，Centreon 和 Nagios 检测主机和服务，从管理主机和服务中收集数据的方式有两种：一种是基于轮询（polling-only）的主动监控模式，另一种基于中断（interrupt-based）的方式。轮询方式是一种基于“请求—响应”同步机制的检测方式，该机制下的主机和服务总是在 Centreon 和 Nagios 中央监控服务器的控制之下。而这种方法的缺陷在于信息的实时性，尤其是错误的实时性。如果 Centreon 中部署的服务数量巨大，或者检测延迟过高时，都会对告警的及时性产生影响。如果轮询间隔太小，那么将产生太多不必要的通信量。如果轮询间隔太大，并且在轮询时顺序不对，当遇到大的灾难性的事件又无法及时发出预警或者告警，这就违背了积极主动的网络管理目的。

而如果采用 SNMP trap 机制，当有异常事件发生时，故障主机或者服务可以立即发送 SNMP trap 消息通知 Centreon 和 Nagios（在这里假设该设备还没有崩溃，并且在被管理设备和管理工作站之间仍有一条可用的通信途径，例如手机短信、微信等即时通信），那么管理

人员就可以及时感知故障的发生。然而，基于 SNMP trap 的故障上报方法也不是没有缺陷的。首先，产生错误以及发送 SNMP trap 消息需要系统资源。如果 SNMP trap 机制必须转发大量的信息，那么被监控主机和服务可能不得不消耗更多的时间和系统资源来产生 SNMP Trap 消息，从而影响了它执行主要的功能。而且，如果几个同类型的 SNMP trap 事件接连发生，那么大量网络带宽可能将被相同的信息所占用。尤其是如果 SNMP trap 是关于网络拥挤问题的时候，事情就会变得特别糟糕。

Centreon 提供了对于 SNMP trap 的管理机制。对于制造商来说，主机或者服务能够产生的每一个 SNMP trap 都对应于一个唯一的 ID，即著名的 OID，OID 的都在设备厂商提供的 MIB 文件里。为了使 SNMP trap 被 Centreon 正确处理和展现，需要事先在 Centreon 中导入 MIB 文件，并且在编辑服务列表中，使其与 MIB 库中每一个 OID 一一对应起来。

2. 声明设备制造商

Centreon 中对于 SNMP trap 的管理始于对设备制造商的管理，因为设备在出厂时，都会由制造商提供对应的 MIB 库，里面定义了各类 OID，以对应设备可能出现的各类状态。详细配置步骤如下。

单击菜单 Configuration→SNMP Traps，再单击左侧树状菜单中的Manufacturer项，可进入设备制造商列表界面，如图 11-59 所示。

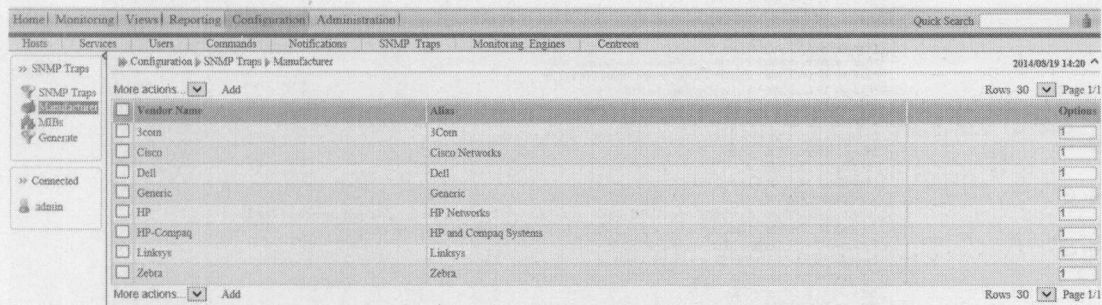


图 11-59 Centreon 中的设备制造商列表

如图 11-59 可见 Centreon 中已经定了一些主流的设备制造商，例如 HP、Cisco 等。接下来单击 Add 链接，可进入新增设备制造商界面。如图 11-60 所示，我们定义了名为“华为”的设备制造商。

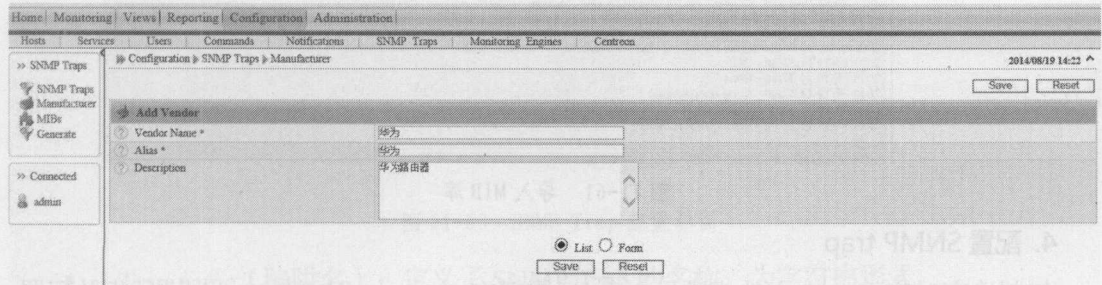


图 11-60 新增设备制造商

3. 导入 MIB 库

Centreon 提供了 MIB 库导入界面，避免了手动输入一个个设备 OID 的麻烦。注意 MIB 库之间如果存在依赖关系，需要逐级逐个导入 MIB，具备最少依赖关系的 MIB 文件应该被最先导入，其次在逐级导入。待 Centreon 处理完毕并构建起 MIB 文件之间的依赖关系后，一套 SNMP trap 告警机制就建立起来了。

单击菜单 Configuration →SNMP Traps，再单击左侧树状菜单中的 MIBS 项，可进入 MIB 库导入界面。如图 11-61 所示，首先选择 Vendor Name（设备制造商），然后选择要导入的 MIB 文件，单击 import（导入）。

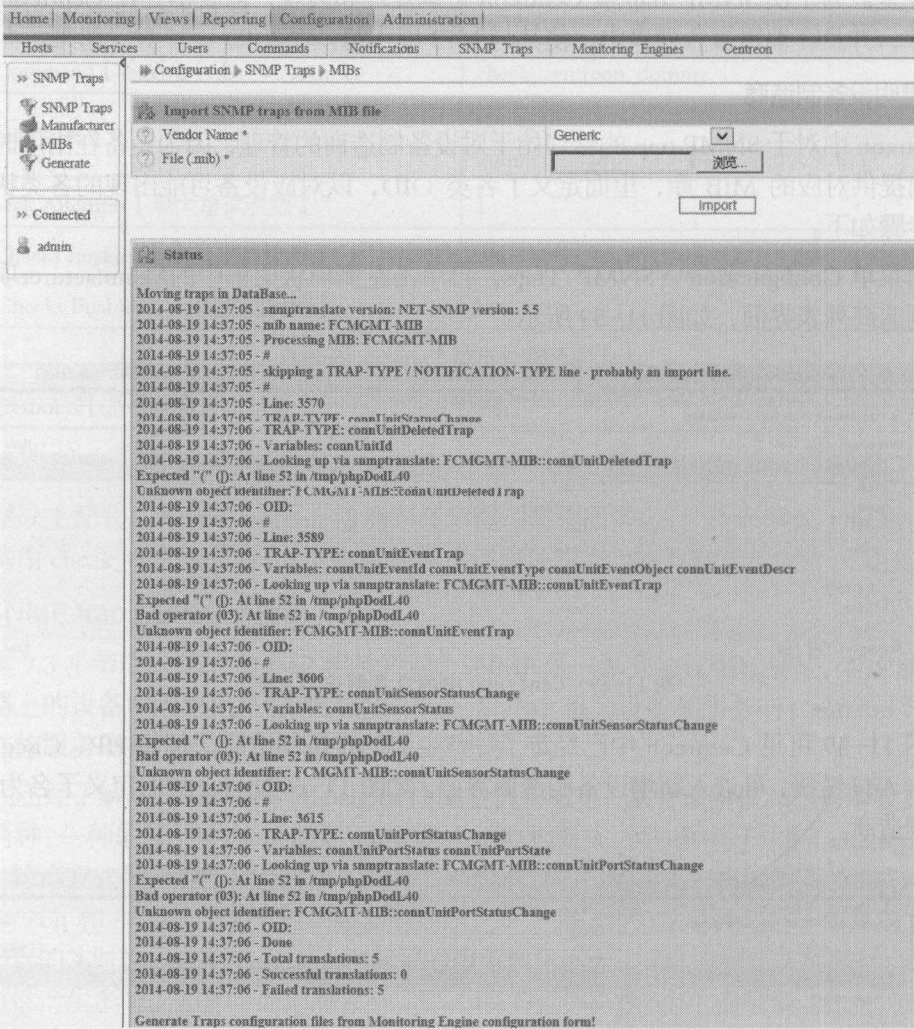


图 11-61 导入 MIB 库

4. 配置 SNMP trap

与制造商对应的 MIB 文件成功导入后，还需要在 Centreon 对 SNMP trap 进行相关配置。单击菜单 Configuration → SNMP Traps，可进入 Centreon 默认的 SNMP Trap 配置界面，如图 11-62 所示。

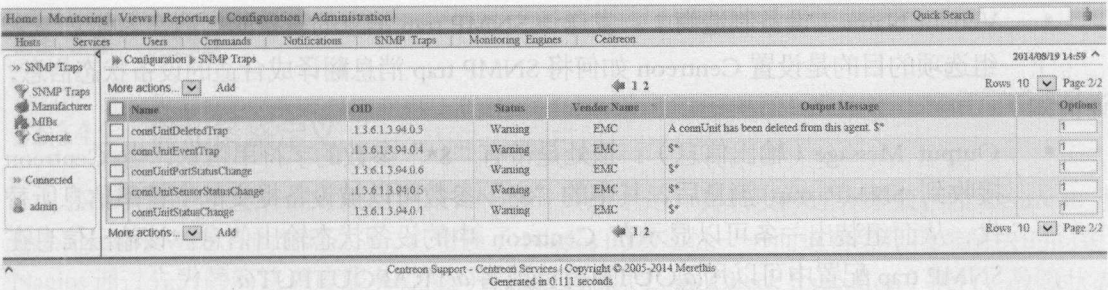


图 11-62 SNMP Trap 列表

单击上图中的 OID 条目，可进入该 SNMP Trap 项的详细配置界面，如图 11-63 所示。

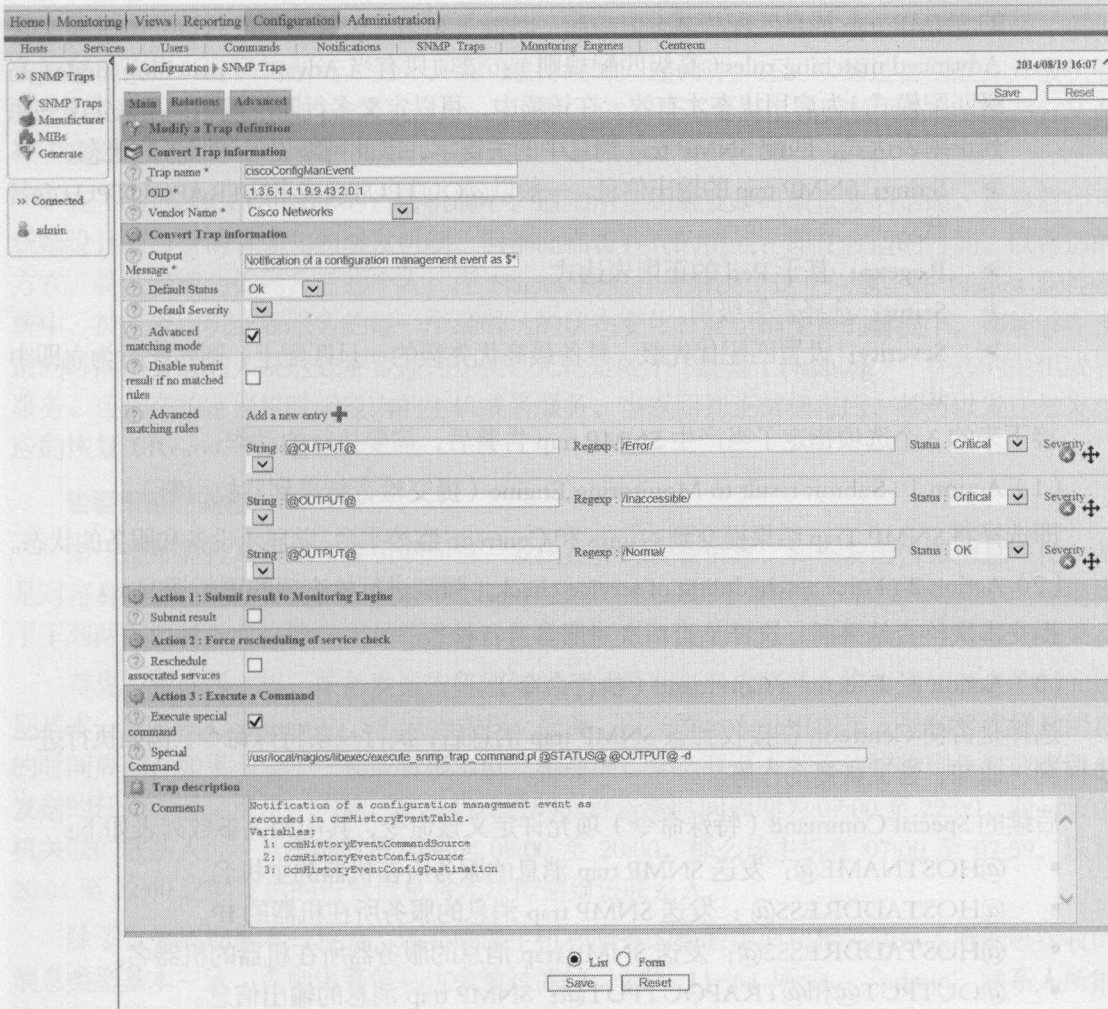


图 11-63 SNMP Trap 配置界面

- Trap name（陷阱名）：定义了 SNMP Trap 的名称，为字符串形式。
- OID：是 SNMP trap 的唯一标识，注意 Centreon 可以区分源于不同设备制造商的重复的 OID。

- Vendor Name (设备制造商) : 发送该 SNMP trap 消息的设备的制造商名称。

下一组选项的目的是设置 Centreon 如何将 SNMP trap 消息翻译成合适的设备状态信息, 详情如下:

- Output Message (输出信息) : 该处是带有 “\$*” 参数的字符串信息, 当 Centreon 接收到 SNMP trap 消息后, 其中的 “\$*” 参数可以被设备提交的字符串信息所替代, 从而组装出一条可以显示在 Centreon 中的设备状态输出信息。该输出信息在 SNMP trap 配置中可以用 @OUTPUT@ 或者 @TRAPOUTPUT@ 替代。
- Default Status (默认状态) : 该 SNMP trap 的默认状态信息, 一般为 “0”, 即 “OK” 状态。
- Advanced matching mode (高级匹配模式) : 在此选择是否启用基于 Perl 正则表达式的 SNMP trap 消息匹配模式。
- Advanced matching rules (高级匹配规则) : 该项只有当 Advanced matching mode (高级匹配模式) 为启用状态才有效。在该项中, 可以定义多行匹配规则, 用来定义一系列正则表达式, 匹配 SNMP trap 消息中的关键字, 以此判断设备和服务的状态。
 - String: SNMP trap 的输出信息, 一般以 @OUTPUT@ 或者 @TRAPOUTPUT@ 替代。
 - Regexp: 基于 Perl 的正则表达式。
 - Status: 选择告警级别。
 - Severity: 设置匹配优先级, 具备最高优先级的一旦匹配上, 则匹配行为立即中止。

接下来的 3 个选项指定了当产生 SNMP trap 告警后, 需要执行的动作:

(1) Action 1 : Submit result to Monitoring Engine (提交检查结果到监控引擎)。

即选择将 SNMP Trap 结果提交到 Nagios 和 Centreon 监控平台, 以显示设备和服务的状态。

(2) Action 2 : Force rescheduling of service check (强制进行检查项的重新调度)。

提交本次检查结果后, 选择是否再次对服务进行检查。

(3) Action 3 : Execute a Command (执行命令)。

选择是否使 Centreon 在接收到该 SNMP trap 消息后, 执行一条特殊命令, 例如执行进一步探测、通知、尝试恢复动作等。

后续的 Special Command (特殊命令) 项允许定义该命令, 其中相关参数列表如下:

- @HOSTNAME@: 发送 SNMP trap 消息的服务所在机器的主机名。
- @HOSTADDRESS@: 发送 SNMP trap 消息的服务所在机器的 IP。
- @HOSTADDRESS2@: 发送 SNMP trap 消息的服务器所在机器的机器名。
- @OUTPUT@ 和 @TRAPOUTPUT@: SNMP trap 消息的输出信息。
- @TIME@: Centreon 监控系统接收到该条 SNMP trap 消息的时间戳。

11.14 通知

11.14.1 通知策略定义

本书介绍到这里，读者就已明白，Nagios 实质上是一套调度和通知系统，而 Centreon 正是 Nagios 的 Web 界面管理工具，它增强了 Nagios 的易用性——而这曾经是很难解决的问题。Nagios 通过先进的算法调度服务检测——如长检测（Long Check Execution）执行、故障状态重试、为避免远程系统产生较高负载的交错检测以及延迟检测等手段，从而管控 Nagios 服务器负载。Nagios 依然会并发运行检测，然后使用信息收集进程（Reaper）从消息队列中获得结果。Nagios 根据状态的变化以及状态的类型（硬状态（Hard）或软状态（Soft））触发通知。通知事件可以转而配置为发送电子邮件、或调用类似企业内部或者公共短信平台（例如中国移动飞信）这样的系统。另外，当问题持续时间过长时，还可以通过告警通知升级机制（Escalation）进行进一步的告警。此外，正常（OK）状态会先过渡到软（Soft）状态，从而减少重要性较低的瞬态问题的通知。

在 Nagios 和 Centreon 的告警机制中，通知消息的触发是一项重要的话题。很多监控系统都设计有告警消息通知的触发机制，但都面临着共同的挑战：即在合适的时间，以合适的方式，将告警消息传到合适的个人，而 Nagios 则很好地解决了这个问题。在 Nagios 监控系统中，被监控主机或者服务的每一个经确认的状态变化，在正确配置的前提下，都能够在合适的时间向合适的人发送数量合适的短信。无论是处于“震荡（Flapping）”状态的主机或者服务，还是在预定停机时间段内的主机或者服务，当返回到正常状态后，都可以发送触发相应的恢复（Recovery）通知。

告警策略的调整

对于运维监控系统的告警机制而言，发出过多的告警信息，和不发出告警信息一样，都是对自身可靠性的一种伤害。但对于 Nagios 来说，这一切不是问题，因为 Nagios 已经提供了下列两类机制，允许管理人员精准地对告警机制进行调节，以达到最佳的告警通知效果。

首先，要确保主机、服务发送告警通知的时间周期与各自联系人接收通知消息的时间周期基本一致。在 Centreon 中，可以为主机、服务，以及联系人分别指定发送和接收通知消息的时间周期，如果两者不一致或者根本就互相冲突，那么联系人就无法接收到主机或者服务发送的任何的通知消息。例如，某主机设置的允许告警时间周期为 07:00 至 22:00，而与该主机关联的联系人允许告警的时间周期为 08:00 至 20:00，那么该主机在 07:00 至 07:59，以及 20:01 至 22:00 之间发送的告警信息将无法传递到该联系人。

除了告警周期基本一致外，还要确保主机和服务的告警类型和相关联系人所能够接收的消息类型基本一致，否则告警信息也会被过滤掉。如图 11-64 所示，“admin”联系人所能接收的告警类型中缺乏“Warning”类型的告警信息，这意味着该联系人无法接收主机或者服务发出的类型为“警告（Warning）”类型的消息，而其他类型的告警信息只要在合适的时间段内，都不会被过滤掉，且能够正常接收。

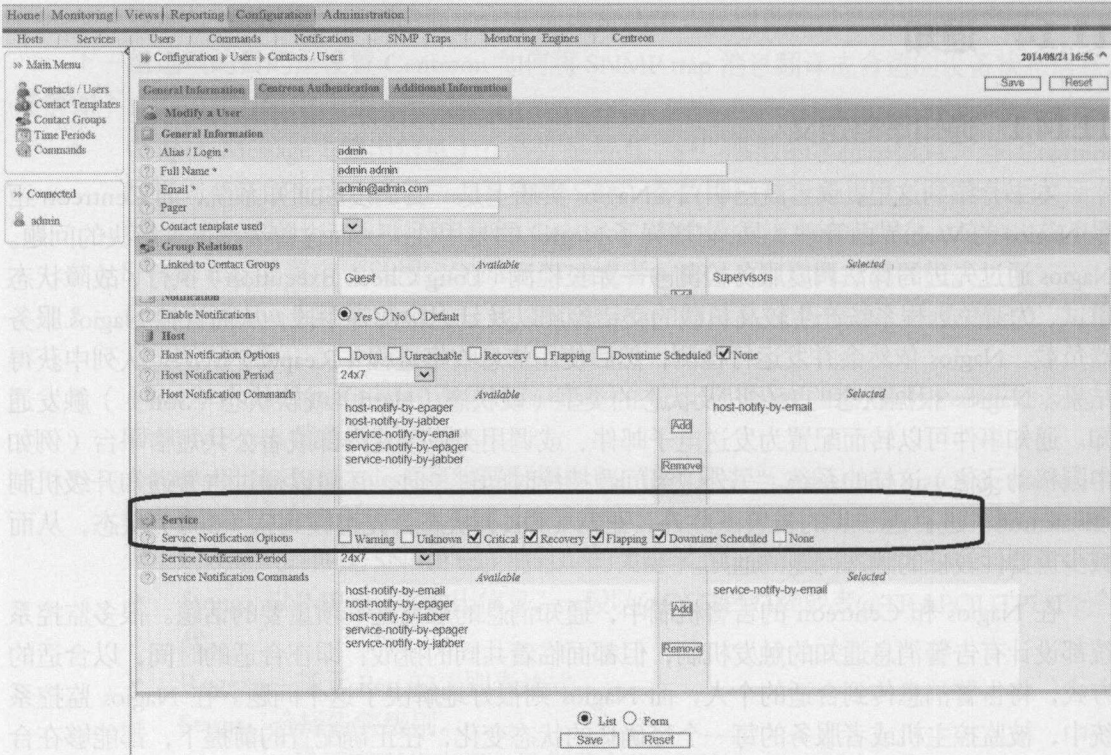


图 11-64 联系人告警消息过滤

其次，要通过为主机和服务设置依赖关系的方式，限制告警信息的数量。Nagios 允许用户定义主机和服务之间的依赖关系来控制主机和服务，遵循如下规则：

- 一个服务可以依赖于一个或者多个其他服务；
- 一个服务也可以依赖于不在同一台主机上的其他服务；
- 通过服务依赖，用户可以使 Nagios 忽略一定次数的通知消息和服务检测；
- 可以为依赖关系指定有效期。

依赖关系在主机和服务之间占很重要的地位。比如服务依赖的情况，很多服务都依赖于主机的可连接性，网页的可访问依赖于 Web 服务器的状态。一般情况下，如果主机已经宕机，它上面的其他服务的检测也会失败。如果主机上的服务由不同的管理员负责，通过 Nagios 进行检测，只需通报主机已经宕机的消息给系统管理员即可，因为即使得到其他相关服务无法访问的消息，其他服务管理员也无力处理这个问题。主机依赖也有相似的状况，比如 A 主机通过 B 主机连接网络提供服务，如果监控到 B 主机停止的话，则不再对 A 主机的故障情况发出告警通知。

如图 11-65 所示，定义了一个较为复杂的检测依赖和通知消息依赖关系图，箭头所指对象为“被依赖的”对象：

服务依赖关系图

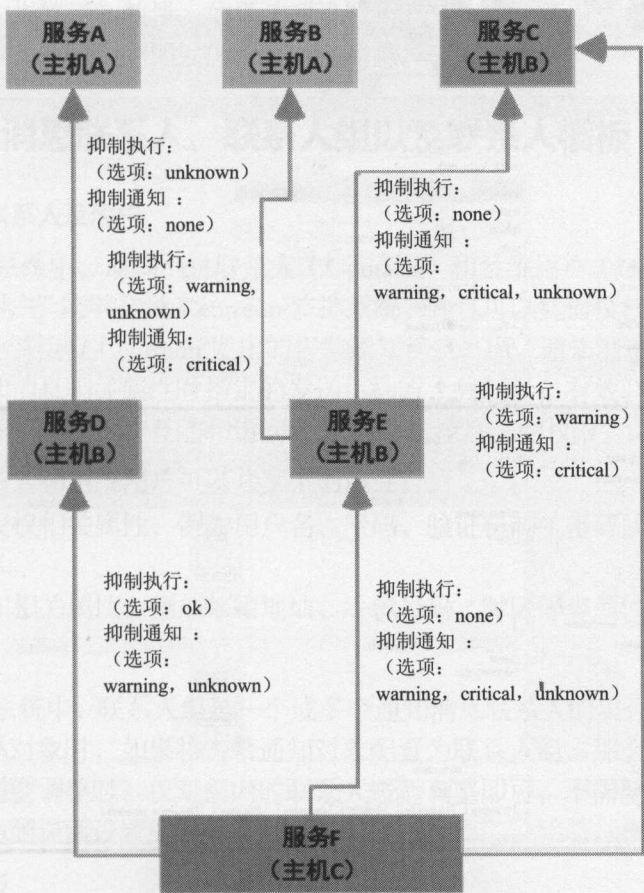


图 11-65 主机和服务之间的依赖关系

如图 11-65 所示，主机 C 上的服务 F 依赖位于主机 B 上的服务 D 和服务 E，同时依赖位于主机 B 上的服务 C，且各自的抑制通知选项各有不同。这意味着如果服务 D 的状态为 warning/unknown，以及（或者）服务 E 的状态为 warning/critical/unknown，以及（或者）服务 C 的状态为 critical，那么 Nagios 将会直接送出服务 D、E、C 各自的告警信息，同时忽略掉 F 的任何告警信息，因为如果没有依赖关系，服务 D、E、C 出现“抑制通知”选项定义的相应状态后，服务 F 肯定会受到影响，并发出告警，而基于预先设计的依赖关系，正好能够避免 Nagios 发出服务 F 的多余告警信息。

11.14.2 为主机和服务配置通知策略

在 Centreon 的 Web 界面中，可以在主机和服务详细编辑页面的第一个选项卡中配置相关通知策略。另外，还可以通过编辑主机模板和服务模板来配置通知策略，以模板方式配置的告警通知策略可以被主机和服务所继承。

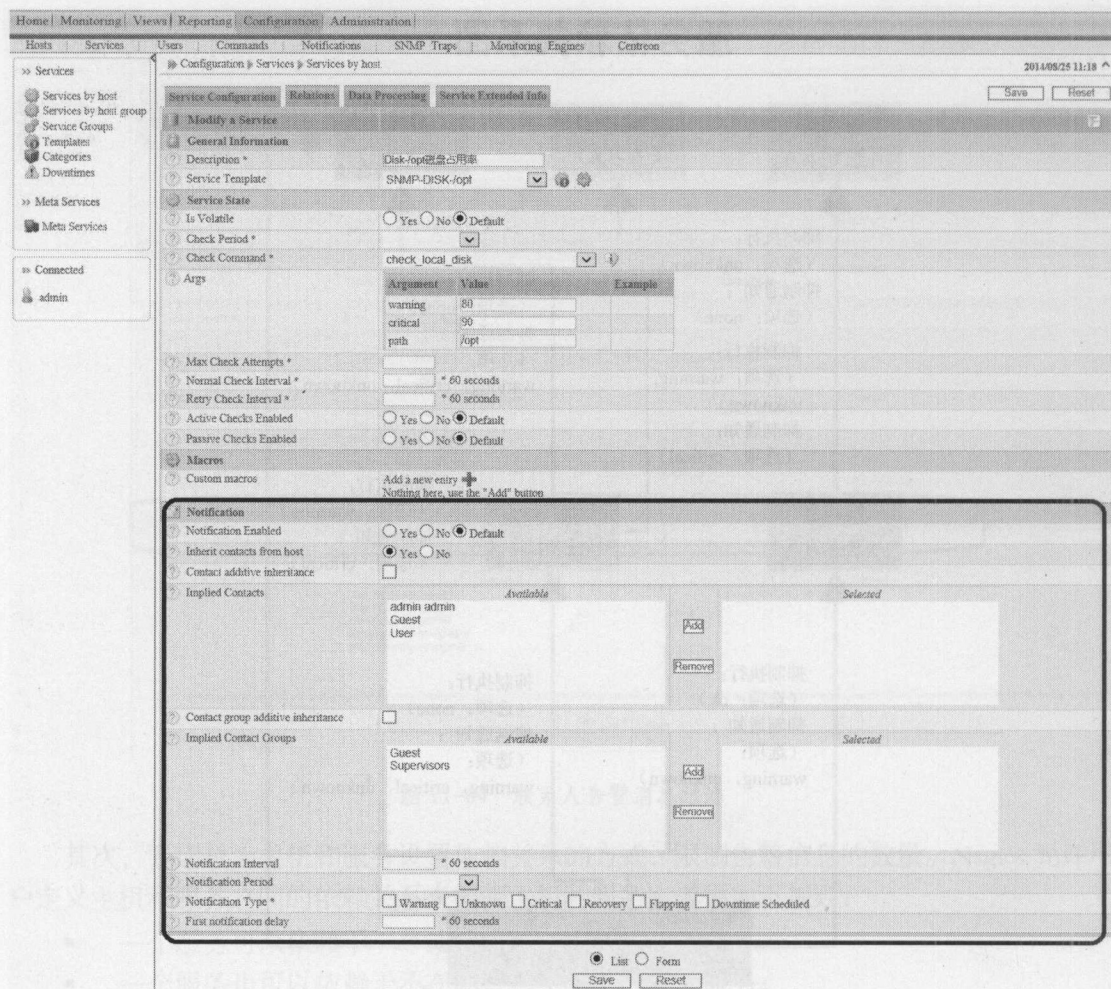


图 11-66 为服务配置通知策略

如图 11-66 所示界面中的“Notification”选项卡里，用户可以启用或者禁用主机或者服务的告警通知功能，为主机或者服务指定不同的告警联系人组，指定发送重复告警消息的次数，以及设定发送告警消息的时间周期，最后，还可以选择特定的能够发送通知消息的状态类型，例如，某服务如果处于 Warning 状态，则不发送通知告警消息，如果处于 Critical 状态，则发送告警消息等。有关检查周期和通知告警时间周期的定义可以通过单击菜单 Configuration Users→ Time Periods，在“时间周期”列表界面（如图 11-67 所示）中进行操作。

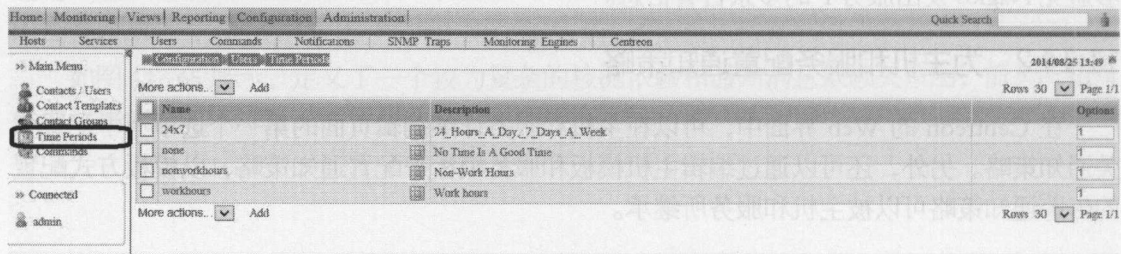


图 11-67 Centreon 中有关“时间周期”的定义页面

图 11-66 中的 First notification delay 项定义了从服务被检测出处于“故障(非 OK 状态)”状态,到发出第一条告警通知消息之间的延迟时间,以分钟为单位。设定该延迟时间有助于 Nagios 降低服务的告警通知数量。如果该项设置为 0,意味着 Nagios 检测到该服务处于非正常状态时,会根据预定规则立即发出告警通知消息。

11.15 通知消息联系人、联系人组以及联系人模板

1. 通知消息联系人及用户

在 Centreon 系统中,通知消息联系人(Contacts)和系统用户(Users)是容易混淆的两个概念。“系统用户”能够登录 Centreon 监控系统,并且可以凭借授权访问监控系统的一系列功能,同时也能够接收 Centreon 发出的告警通知消息。而“通知消息联系人”是在为主机或者服务配置通知消息相关属性时指定的发送目标方,只要具备有效联系方式即可,例如邮箱、手机号等,并不一定具有登陆和访问 Centreon 监控系统的权限。

通知消息联系人和系统用户可以共享下列属性:

- 验证和授权相关属性,例如用户名、密码、验证机制(密码验证、或者通过 LDAP 验证)等。
- 消息通知相关属性,例如邮箱地址、手机号码、时区等等属性。

2. 联系人组

在 Centreon 系统中,联系人组是一个或多个通知消息联系人的集合。在为主机或者服务配置通知消息联系对象时,如果将告警通知对象配置为联系人组,那么当遇到主机或者服务的告警通知对象需要调整时,仅对组内的联系人进行调整即可,不需要调整主机或者服务的相应属性,从而达到灵活设置的目的。

3. 联系人模板

与主机模板和服务模板类似,在 Centreon 监控系统中,同样可以定义联系人模板。在联系人模板中,可以设置名称、邮箱地址、手机号码,以及所能够接收的主机通知或者服务通知类型、通知方式等属性。在实际定义联系人时,可以选择相应的模板用以继承模板中已定义的这一系列属性。

实际上,在定义通知消息联系人之前,定义相应的联系人模板是一个好的实践,这样有利于将与联系人相关的属性规范化、一致化,有利于以后的联系人管理。

下面我们来配置通知消息联系人/用户。

通过单击菜单 Configuration→Users,可进入通知消息联系人/用户配置页面,如图 11-68 所示。

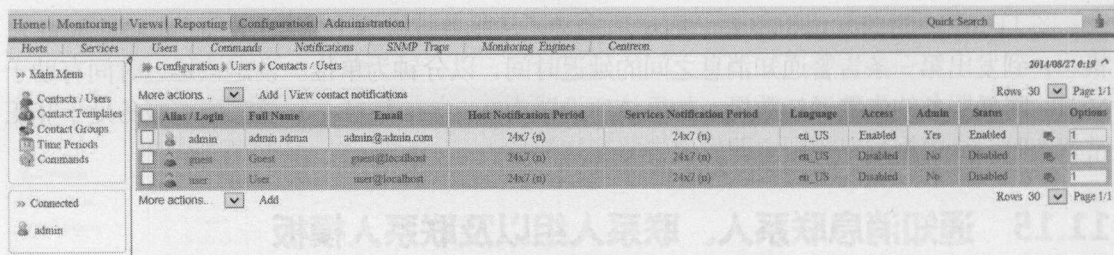


图 11-68 通知消息联系人/用户列表界面

单击图 11-68 中的 Add 链接，可进入新增通知消息联系人/用户界面；单击列表中的联系人名称，可进入如图 11-69 所示的编辑界面：

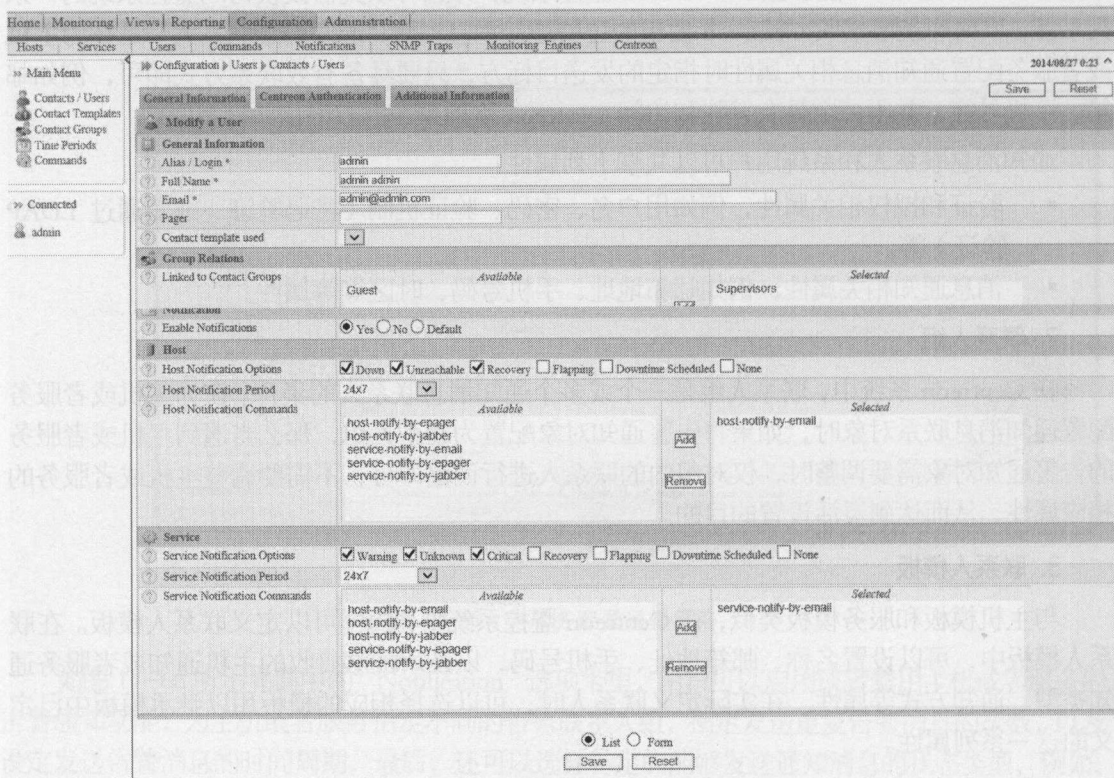


图 11-69 通知消息联系人通用信息编辑界面

General Information（通用信息）选项卡

如图 11-69 所示的与联系人/用户常用信息相关的配置项介绍如下：

- Alias / Login（登录名）和 Full Name（联系人/用户全称）：分别指定联系人/用户的登录名称以及全称。前者是登录 Centreon 监控系统的登录 ID，后者是该联系人的全名。
- Email（邮箱）和 Pager（寻呼号码/移动终端号码）：可以分别设置联系人/用户的邮箱地址和寻呼机、手机等移动终端的号码。Centreon 可以按照此处定义的多种联系方式将告警消息通知到联系人/用户。
- Contact template used（所选用的联系人/用户模板）：可以在此选择之前已经在

Centreon 系统中定义好的联系人/用户模板。

- **Linked to Contact Groups (关联联系人/用户组)：**此处可以将联系人/用户与联系人组关联起来，在此处所编辑的关联关系同时可以体现在该联系人组的编辑界面中。
- **Notification (告警通知)、Host (主机)、Services (服务) 选项组：**该选项组配置了该联系人/用户接收主机告警通知和服务告警通知的相关全局选项，例如是否接收告警通知消息、接收何种类型的告警通知消息（警告、紧急、恢复、正常等），以及采用何种类型的接收方式（邮件、寻呼机、手机短信等）等。

注意：主机或者服务告警信息是否发出的相关配置（参看图 11-44 和图 11-53）和联系人是否能够正常接收告警信息的相关配置（参看图 11-69）是两类不同的配置项。两者之间需要存在匹配，才能够使告警消息精准地发送给联系人/用户（至于联系人/用户是否查看以及通信链路是否正常不在本书讨论范围内）。因此主机、服务、联系人/用户三者之间的告警通知消息的配置，例如是否发出（接收）告警信息、告警周期、告警类型等选项应尽可能一致，或者不存在明显的冲突。

如图 11-70 所示为 Centreon Authentication (Centreon 验证与授权) 选项卡。

图 11-70 通知消息联系人身份验证与授权界面

- **Reach Centreon Front-end (访问 Centreon 前端界面)：**该选项允许通知消息联系人/用户访问 Centreon 的 Web 用户界面。前面我们提到过，Centreon 中的通知消息联系人和系统用户是两个不同的概念，前者并不一定具备访问 Centreon 系统的权限，而该选项正是用来启用该选项。该选项默认为启用状态。
- **Password (密码) 和 Confirm Password (确认密码) 选项：**可以输入联系人/用户密码。
- **Default Language (默认语言)：**可以选择联系人/用户登录系统后默认使用的语言，修改此项可立即生效，不需要重新登录 Centreon 系统。
- **Admin (是否管理员)：**该选项用以指定联系人/用户是否为 Centreon 监控系统的管理员用户。如果启用该选项，那么 Centreon 系统中定义的权限访问控制选项将不会对该联系人/用户生效，后者将直接具备系统的管理员权限。

- Autologin Key（自动登录秘钥）：定义了联系人/用户自动登录 Centreon 监控系统所使用的密钥，如果使用了单点登录系统登录 Centreon，该令牌就会用到。
- Authentication Source（认证源）：在该处可以选择联系人/用户登录 Centreon 系统的认证方式，是基于 MySQL 数据库的用户名、密码式认证，还是基于 LDAP 目录的认证，默认为基于用户名和密码的认证方式。
- Access lists（访问列表）：在 Centreon 监控系统中，将用户界面的各个功能划分成不同的资源，通过对于访问控制需求的配置，设置一些界面可以被访问，一些界面访问需求可以被拒绝，这就形成了 ACL（Access ControlList，访问控制列表）。因此，该选项是用来配置联系人/用户访问 Centreon 监控系统的访问控制列表。但如果联系人/用户具备管理员权限，那么会自动具备访问全部 Centreon 监控系统 Web 界面的权限，该控制列表将不会生效。

11.16 Commands 通知命令

通过单击菜单 Configuration → Users，再单击左侧竖状菜单中的 Commands 链接，可进入通知命令列表界面，如图 11-71 所示。

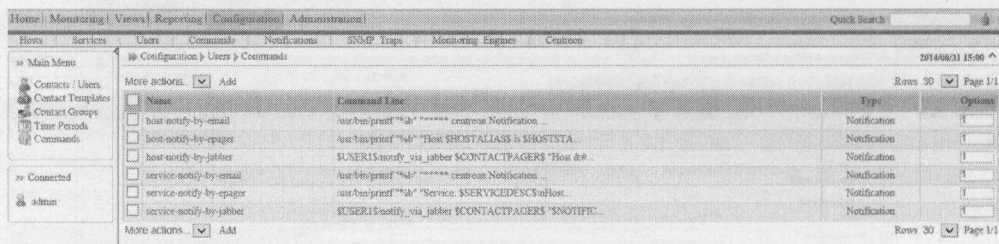


图 11-71 通知命令列表界面

单击上图中的通知命令名，可进入命令编辑界面，如图 11-72 所示。

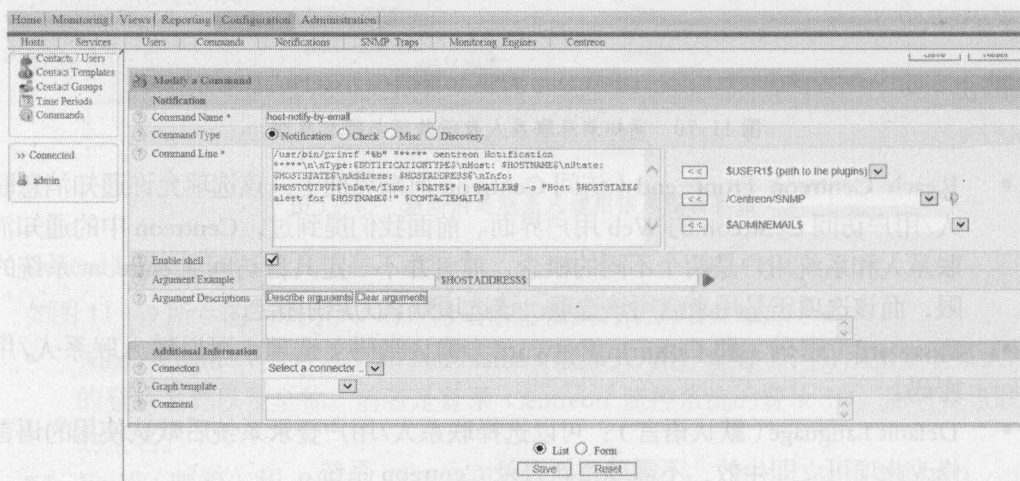


图 11-72 通知命令编辑界面

在上图中的命令编辑界面中, Command Line (命令行) 文本框中显示的是通知命令的详细内容, 几乎所有的变量都是通过宏 (参看 11.4 小节) 来实现的, 当然也可以固定的文本来替换宏, 例如用一个固定的联系人邮箱来替换 \$CONTACTEMAIL\$ 宏, 这样一来所有的告警信息都将会发往此邮箱。

Centreon 系统对于告警通知消息管理的灵活性在于, 可以运用尽可能多的方式, 例如邮件、短信、声音、颜色变化等多种感知手段来传递告警信息, 只要这些手段能够以命令行方式驱动。在移动互联网大行其道的今天, 实现这一点并不困难。

11.17 Escalation-告警通知的升级

Nagios 提供了告警通知的扩展功能, 可以利用该功能对告警通知的次数、通知对象、通知间隔、通知消息类型等行为进行限制, 这就是 Nagios 的告警通知升级功能 (Escalation)。Nagios 之所以引入了告警通知升级功能, 而非简单地送出告警信息, 目的是为了满足不同运维工作中的如下需求:

- 自动将影响较大的告警信息传递给运维团队中领导层, 或者较高职位的人, 以便于后者收到第一手信息并及时处理。
- 如果一线监控人员未能注意到告警信息, 或者接收到告警信息后未能及时处理, 那么系统可以将告警信息及时传递给二线支持团队。
- 如果第一种告警通知方式不可用或者无效, 例如邮箱通知无效的情况下, 使用手机短信等更加快速及时的通知手段发送告警通知。

在日常运维过程中, 很多情况下会根据故障的持续时间、影响范围、受关注程度以及严重程度等维度, 对故障进行定级, 例如将故障定位一类、二类、三类、四类等级别。一般来说, 一类故障是最为严重的故障, 而四类故障是最为轻微的故障。而故障的级别是根据每项服务, 每个产品而定的, 并没有统一的标准, 并不是说所有的业务故障都是一类或四类故障, 要根据业务的特质进行判断, 例如将某项核心业务的所有故障告警都划分为一类, 而其余的非核心业务的所有告警划分为二等等。

举个最典型的电子商务例子, 对于电商网站来说, 核心业务是交易, 因此对于交易速度和交易质量不仅要进行监控, 还要根据监控告警的影响范围、影响时间、影响用户数量等条件设定不同的告警级别。某项交易告警的初期, 可能会有个别用户投诉, 此时为四类或者三类故障, 之后如果告警持续时间较长, 继而引发大量投诉, 告警就必须升级为一类故障。

不同的告警故障应该通知到不同级别的员工。对于四类或者三类故障, 监控平台会将告警信息通知到一线技术支持人员, 由一线技术支持人员响应故障、排查问题、联系投诉用户并分析问题原因, 直至解决问题并消除告警信息。在此过程中, 是很普通的故障响应与解决流程。但是一旦故障持续未能解决, 比如超过了 20 分钟, 发现还未解决的话, 那么故障的性质就会发生变化, 就要启用升级机制, 将故障升级为二类或者一类故障, 将告警信息传递给二线技术支持人员, 直到负责运维的副总裁, 此时需要给副总讲明, 这个故障我们已经处理了一段时间, 至今无法知道什么时候能够解决问题。这个时候就会有更多的人知道发生了故障, 会有更多的人参与、调动更多的资源来响应并解决故障。因为一类故障停留在一线技术支持人员手上, 可能无法最快解决, 也无法调动公司其他资源来处理用户投诉等业务问题。

而故障告警升级后，二线技术支持人员或者更高层的管理者们就会参与进来，调度各类资源，协调整个故障处理，并进行客户公关。现代企业在执行业务流程时需要各个部门全方位协作处理各类问题，而非只有一线技术支持人员独自面对问题，从这个角度来说，为监控平台设计合适的告警通知升级机制是非常有必要的。

一般地，告警通知的升级会在一定数量的普通告警信息通知之后而触发，在 Nagios 为某一告警项发送了一定数量的告警通知信息之后，如果该告警项仍未得到处理（例如从故障状态恢复为正常），那么将由告警通知的升级机制来接管该告警项的后续告警动作，按照预定的升级规则来发送告警消息。

如图 11-73 所示，显示了两类不同的告警通知升级机制：

- (1) 某个监控项被设定为如下策略，如果产生告警信息，就每隔 2 小时联系告警通知组 A；
- (2) 待到第 3 次告警消息的发出时，将会触发首次告警通知升级机制，规定每 1 小时联系告警通知组 B；
- (3) 如果该告警仍未得到处理，那么待轮到第 7 次告警消息发出时，将会触发第 2 类升级机制，在每小时联系告警通知组 B 的同时，进而需要增加联系告警通知组 C。

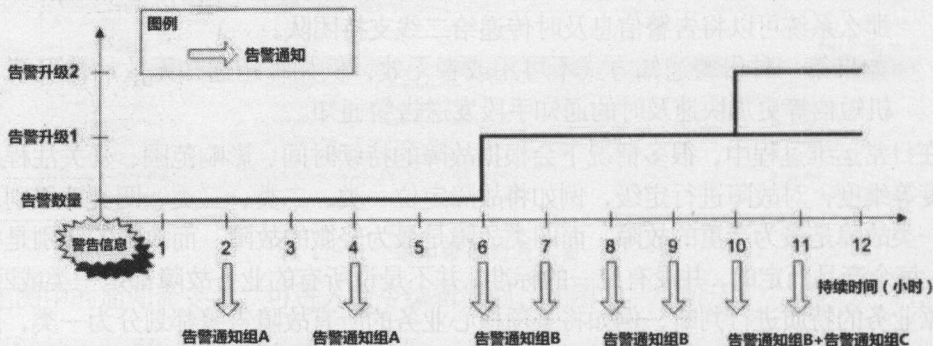


图 11-73 告警通知升级示例

通过选择菜单 Configuration→Notifications，单击左侧竖状菜单里的 Escalations 链接，可进入告警通知升级列表界面，单击“add”链接或者单击告警通知升级项名，可进入告警通知升级项编辑界面，如图 11-74 所示。

- Escalation Name（升级配置项名）：是为该升级配置项指定的名称，具备唯一性。可以在告警通知升级列表中定义多个升级配置项，以制定不同的告警升级策略，注意策略之间应尽可能保持时间段、告警类型、联系人等一致，否则配置项可能被视为无效。
- First Notification（首次通知）：指定了从第 n 次告警通知消息被送出时，该告警通知升级配置项即被触发。注意前面提到过告警通知升级一般是随着前几次普通的告警通知被送出之后再触发的，此处即填写升级机制触发时，告警通知应该是第几次被送出。例如，要求某项服务监控项在连续发出 3 次告警通知后，如果仍未恢复正常，即启动升级机制，那么此处应该填写 4。

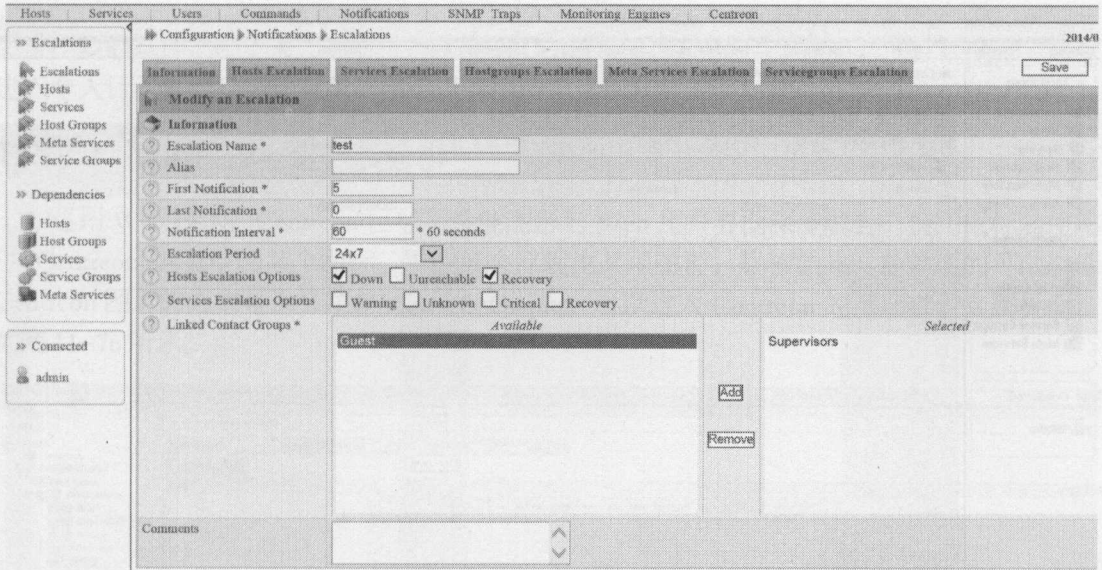


图 11-74 告警通知升级项编辑界面

- Last Notification（末次通知）：此处填写从第 n 次告警通知消息被送出后，该告警通知升级配置项即处于失效状态，是与“首次通知”相对应的配置项。注意如果此处填写 0，表明该配置项永远不失效，直到告警恢复。
- Notification Interval（通知间隔）：此处填写告警通知间隔，单位是分钟。注意如果此处填写 0，Nagios 将在首个告警通知升级项生效后，仅仅发出 1 条告警通知消息，此后再也不发任何告警通知，除非相关故障恢复后再次告警。这一特性对于避免 Nagios 发出过多的告警通知短信很有必要。如果想要 Nagios 在遇到故障后仅发出 1 条告警短信，该项可以设置为 0。
- Escalation Period（升级周期）：此处指定告警通知升级配置项生效的时间段。
- Hosts Escalation Options（主机升级适用的告警通知类型）和 Services Escalation Options（服务升级适用的告警通知类型）：这两项用来选择各自适用于主机和服务通知升级的告警事件类型。
- Linked Contact Groups（关联联系人组）：此处可以选择该告警升级配置项所关联的联系人组。当告警通知升级机制触发后，可以发告警信息给联系人组中相关的联系人。注意此处至少选择 1 项联系人组，否则该配置项无效。

例如，在如图 11-以及图 11-所示的配置项中，配置了名为 test 的告警通知升级配置项。当关联主机 localhostcopy 的状态处于 Down（宕机）状态，如果该状态一直持续到第 4 条普通告警通知消息发出后，仍旧没有恢复，那么轮到下一次（第 5 次）通知告警时，将会触发该告警升级机制，属于 Supervisors 联系人组内的相应人员即会收到告警信息，如图 11-75 所示。

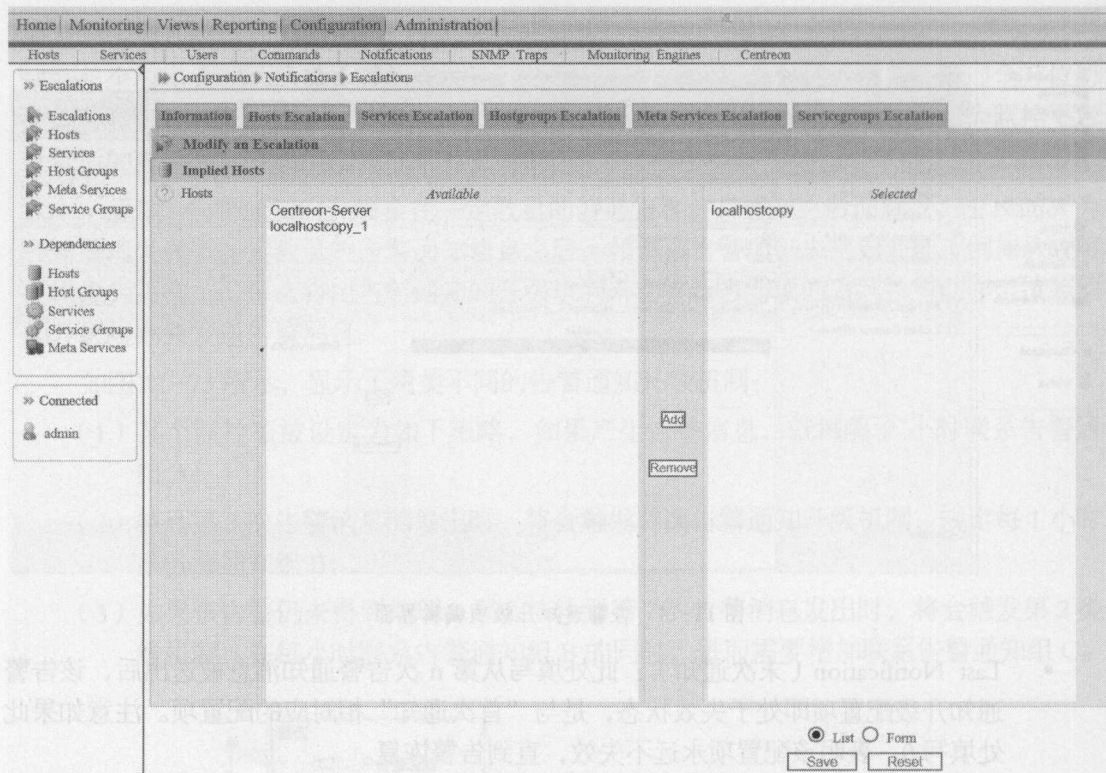


图 11-75 告警通知升级项关联主机

11.18 性能图形

11.18.1 相关定义

性能图形（Performance Graph）是 Centreon 监控系统基于监控项的历史性能数据而自动产生的图形化的数据视图。在 7.5.2 小节和 7.5.3 小节中我们了解到，Centreon 和 Nagios 借助于检测器和检测插件从被监控端采集并存入后台数据库的监控项返回信息中，除了被检测对象的告警信息外，还可以包括符合预定义格式的性能数据。而这些性能数据经由 Centreon 的自动绘图程序绘制之后，就形成了可视化的数据，在 Centreon 的 Web 用户界面中以 PNG 格式图片的形式展示。

Centreon 对于性能图形的展示一般用曲线表示，每条曲线可以代表一项性能数据。如果监控项含有多种性能数据的话，该监控项的性能图形就会包含多条曲线。例如，某个 Oracle 数据库表空间的监控项中，包含有数据库每个表空间的占用率性能数据，那么在 Centreon 为该表空间监控项绘制的性能图形中，就会含有多条曲线，每条曲线代表一项表空间占用率走势。

Centreon 中的性能图形可供用来对监控项的历史告警信息和历史性能信息进行深入的可视化分析。抛开枯燥无味的数字和时间，进入生动鲜明的图形世界中，可以让管理人员轻松了解系统过去的状态，甚至据以预测未来的走势。除了对某项指标作趋势分析外，还可以结合多项指标进行交叉对比分析，进而确定报警根源，可以使管理人员对于系统的健康程度有

更进一步的掌握。接下来，本书将为您介绍 Centreon 监控系统中丰富多彩的性能图形，带您进入令人目不暇接且生动有趣的图形世界中。

11.18.2 查看图形与进一步分析

运用 Web 技术，Centreon 已经将性能图形与 Web 用户界面深度融合，使用鼠标随便略过 Centreon 网页中的某个图标，可能会浮现出某项图形，尽管如此，最简便的可以查看 Centreon 图形的方式是选择菜单 Views → Graphs，可以进入 Centreon 的图形查看专属界面，如图 11-76 所示。

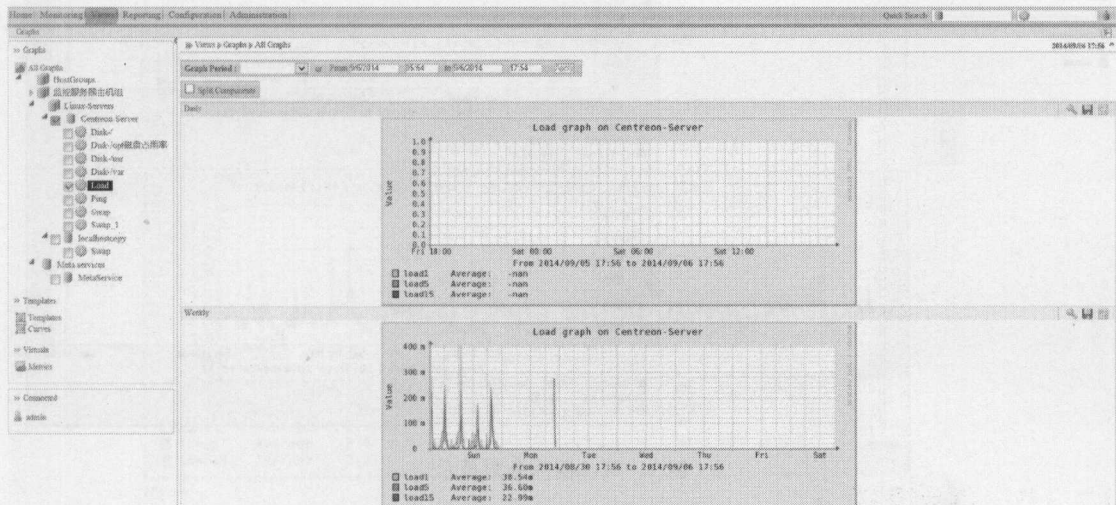


图 11-76 Centreon 的性能图形查看页面

通过单击图 11-76 左侧的竖状菜单，用户可以选择某台主机下的一个或者多个服务来查看其性能图形，还可以选择不同主机下的服务。Centreon 默认根据最近 12 个小时内的性能数据绘制图形，这样有助于用户观察主机和服务近期的行为。需要提醒的是，如果主机项或者服务项仅仅能返回状态，不能返回性能数据的话，Centreon 是无法为它们凭空绘制出性能图形的。通过选择图 11-76 中的 Graph Period 下拉列表，或者选择想要的时间段，用户可以命令 Centreon 基于不同时间段内的性能数据来绘制图形，这样更有利于监视主机或者服务的状态。

在图 11-76 中，如果选择查看 Load 项，默认在一张图片里绘制 Load1、Load5 和 Load15 共 3 项参数的图形，以便于对照查看。但是选择 Split Components 复选框，则 Centreon 会将包含有多个参数的图形分割开来，按照每个参数一张图片的方式显示，如图 11-77 所示。

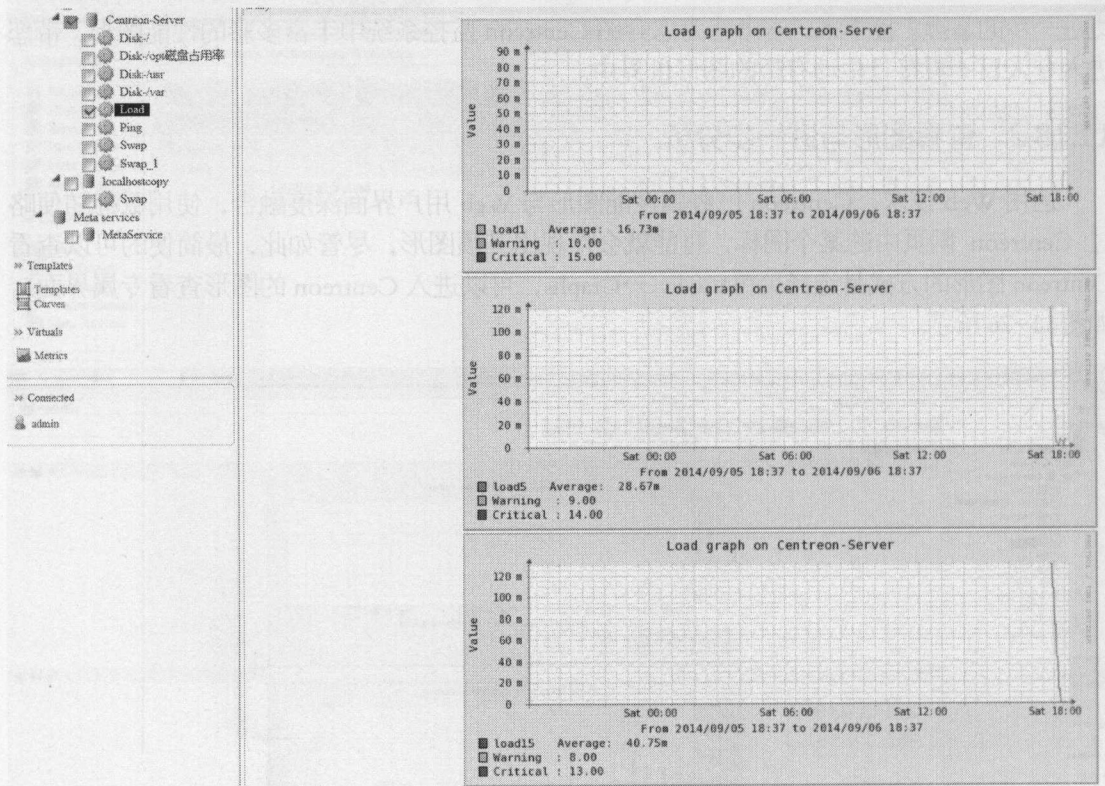


图 11-77 图片分割显示

1. 从图形中观测性能趋势

一直以来，从 Centreon 的图形中观测主机或者服务性能的趋势是用户最想使用的功能，但基于短期的数据，例如几个星期或者几个月的数据，很难达到目的。基于此，Centreon 在一张 Web 页面中提供了 4 类数据供用户观测监控项的性能趋势，分别是基于日、周、月、年来统计数据。通过鼠标单击任意一张性能图片，用户可以进入该页面，如图 11-78 所示。

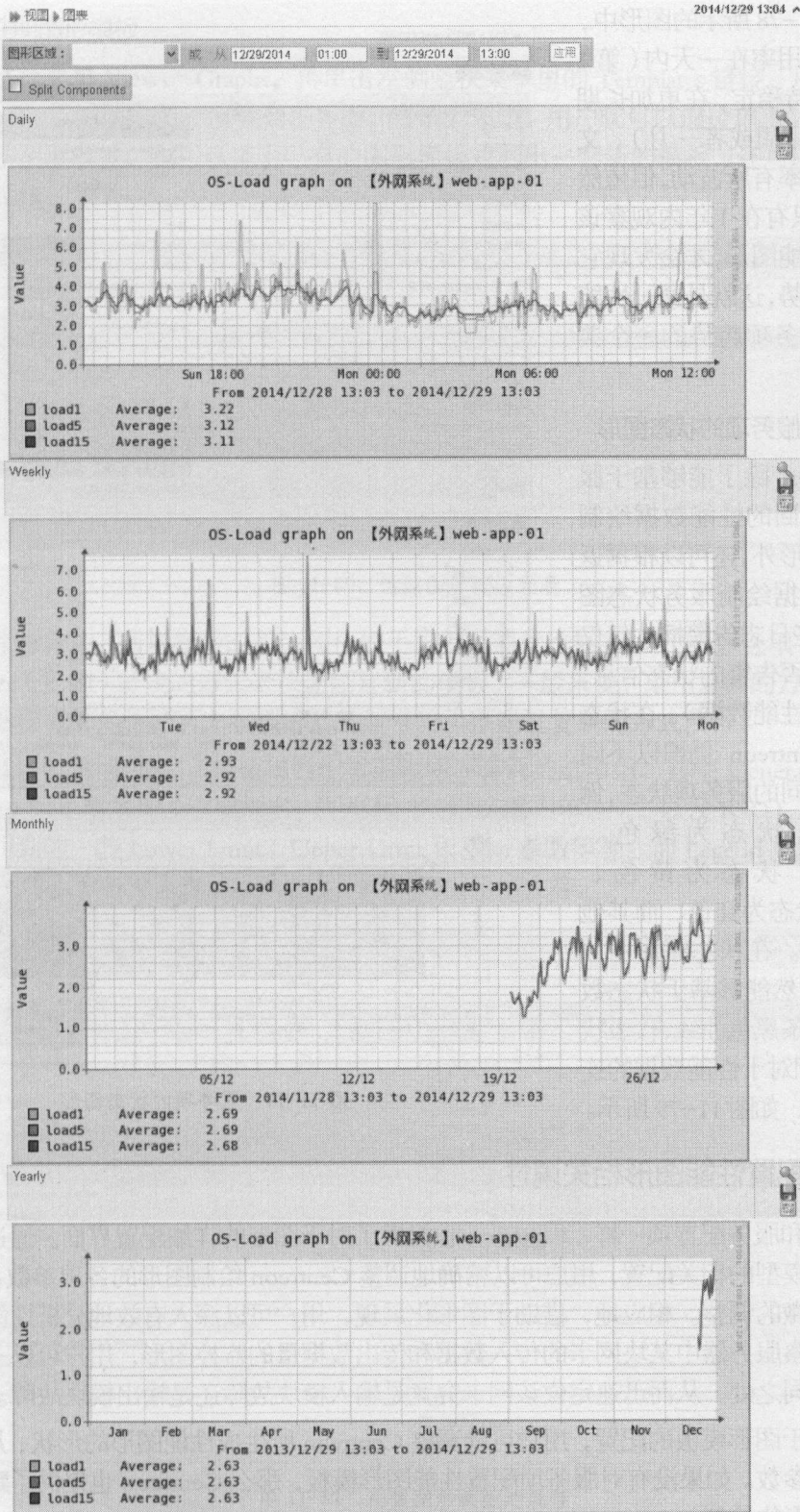


图 11-78 长期性能趋势图

在图 11-78 所示的图形中，文件系统使用率在一天内（第 1 张图片）保持稳定，在更加长期的范围内（一周或者一月），文件系统使用率有所波动，但依然保持稳定。只有在 1 年内观察该项服务的性能图形，才会发现它是呈上升趋势，这就是通过性能图形判断服务项趋势的一个典型例子。

2. 查看服务项的状态图形

Centreon 除了能够基于服务检测项返回的性能数据绘制性能趋势图形外，还可以根据返回的状态数据绘制服务状态图形（例如某些日志告警插件仅仅能够返回是否告警的状态信息，而无法返回性能数据）。在状态图形中，Centreon 仍旧以不同颜色显示不同的服务项状态，例如“OK”状态为绿色、“Warning”状态为黄色、“Critical”状态为红色、而其他状态为白色。在状态图形中，Centreon 依然能够基于状态数据绘制出一条黑色曲线，代表状态的突变（相对于性能数据的连续性而言），如图 11-79 所示。

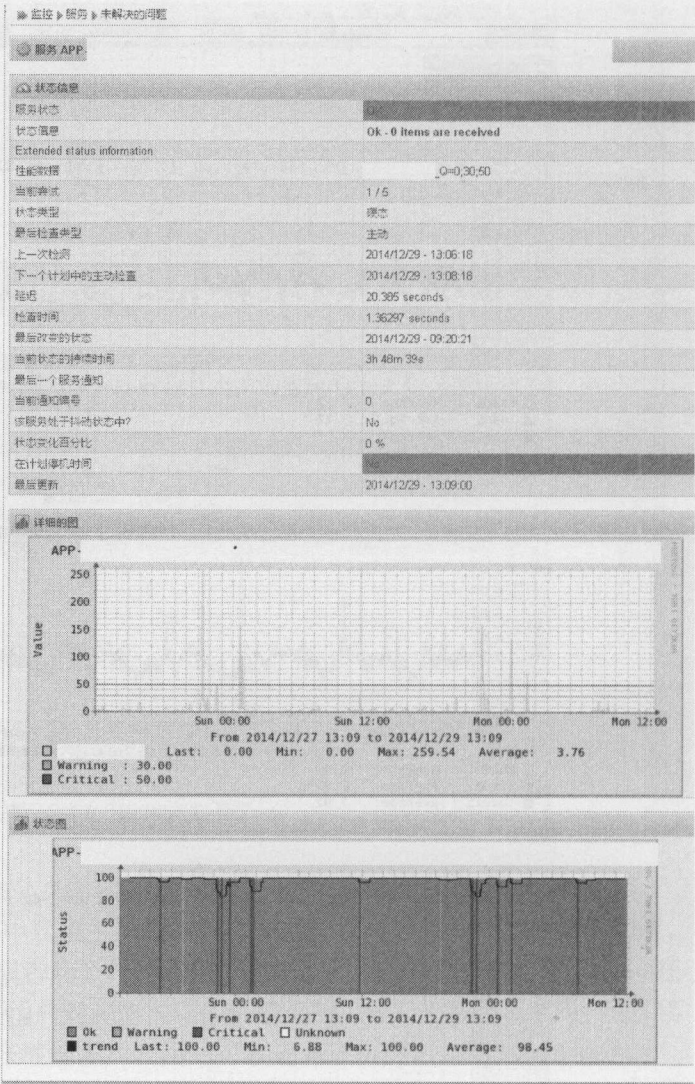


图 11-79 服务项的状态图形

11.18.3 配置性能图形相关属性

与主机和服务配置项一样，Centreon 也提供了对于图形的详细配置界面。通过对于图形模板和曲线模型的相关配置，用户可以精确地调整 Centreon 绘制图形的各项参数，对生成的图形进行细微的调整。相应地，借助于图形化展现，用户可以深入有效地分析性能数据。例如，通过观察服务器中某块网卡的传入数据和传出数据量的监控图形，用户可以一眼看出两者之间的不同之处，从而迅速定位该网卡究竟是输入模块故障还是输出模块故障。

通过对于图形模板的配置，用户可以调整 Centreon 所生成性能图形的形状、尺寸、颜色以及范围等参数。如果没有对服务项配置性能图形模板，那么 Centreon 也提供了默认的模式以供性能图形的绘制。

配置性能图形模板

通过选择菜单 Views→Graphs，再单击左侧竖状菜单里的 Templates 链接，用户可以进入如图 11-80 所示的性能图形模板列表界面。在该列表中，用户既可以通过单击 Add 链接的方式新增一项图形模板，也可以基于已有的图形模板复制出一套新的模板，还可以对既有的模板作删除操作：

Home | Monitoring | Views | Reporting | Configuration | Administration

Quick Search

2014/09/06 21:25

Views > Graphs > Templates

More actions... Add

Rows 30 Page 1/1

Name	Description	Base	Split Components	Options
<input type="checkbox"/> CPU	Processor Use	1000	Yes	<input type="text" value="1"/>
<input type="checkbox"/> Default Graph	Value	1000	No	<input type="text" value="1"/>
<input type="checkbox"/> Latency	Latency	1000	No	<input type="text" value="1"/>
<input type="checkbox"/> Load Average	Load Average	1000	No	<input type="text" value="1"/>
<input type="checkbox"/> Memory	Memory	1024	No	<input type="text" value="1"/>
<input type="checkbox"/> Storage	Storage	1024	No	<input type="text" value="1"/>
<input type="checkbox"/> Traffic	Traffic	1000	No	<input type="text" value="1"/>
<input type="checkbox"/> Uptime	Uptime	1000	No	<input type="text" value="1"/>

More actions... Add

Rows 30 Page 1/1

图 11-80 性能图形模板列表

在日常操作中，新增模板的最为便捷的手段就是通过选择 More（更多选项）下拉列表中的 Duplicate（复制）链接的方式来快速创建新的模板。一般来说，采用复制的方式新增模板，只需修改新模板的名称即可，模板的其他属性均可以通过复制的方式获得。

如图 11-81 所示的图形模板属性配置面板中，有两类选项组，其中 General Information（通用信息）组主要用来设置模板名、图形尺寸（以像素为单位）、以及 RRD 工具在绘制图形时需要用到的一些 Lower Limit、Upper Limit 和 Base 参数等等。而 Legend（图例）选项组则规定了图片中图表的颜色相关属性以及曲线的显示属性等。

Home | Monitoring | Views | Reporting | Configuration | Administration

2014/09/07 12:53

Views > Graphs > Templates

Modify a Graph Template

General Information

Template Name * CPU

Vertical Label * Processor Use

Width * 550 px

Height * 140 px

Lower Limit 0

Upper Limit 110 Size to max ☐

Base 1000

Legend

Grid background color Modify

Text color Modify

Arrow color #FF0000 Modify

Top color Modify

Bottom color Modify

Split Components ☒

Scale Graph Values ☒

Default Centreon Graph Template ☐

Comments

List Form

Save Delete

图 11-81 配置图形模板的属性

General Information（通用信息）选项组

- **Template Name（模板名）**：设定图形模板的名称，可以与一项服务名（例如 check_cpu、check_load 等）、服务模板名或者一项检测命令（例如 check_snmp）相关联。该模板名并不会在绘制完毕的图形中出现。
- **Vertical Label（纵轴名称）**：指的是在绘制好的性能图形中出现的坐标纵轴的名称，显示纵轴的标题，例如 CPU 使用率、表空间使用率等文字标题。
- **Width（宽）和 Height（高）**：定义了图片的尺寸，以像素为单位。
- **Lower Limit（下限值）和 Upper Limit（上限值）**：定义了图形中垂直坐标的限制值，可以选择 Size to max（最大上限）复选框，使坐标的纵轴达到相关性能数据值的上限。
- **Base（计算基准）**：用来设定计算单位的基准值。例如，如果性能数据的单位为伏特（电压值），那么选择基准值为 1000 即可（1 千伏=1000 伏特），如果性能数据的单位为字节，那么选择基准值就应该是 1024（1 千字节=1024 字节）。

Legend（图例）选项组

- **Color（颜色）**：允许用户设定 Centreon 生成性能图片时使用的颜色、包括背景色、文本色、曲线色等等。既可以直接输入颜色值，也可以单击 Modify 链接选取合适的颜色。
- **Split Components（分割曲线）**：允许用户设定每张性能图片上只允许绘制一条性能曲线，而非多条，这样可以提高可读性。
- **Scale Graph Values（设定图像值）**：该选项允许图片自动调整下限值或者上限值以适应性能曲线的显示。但同时这种自动调整行为也受到 Lower Limit（下限值）和 Upper Limit（上限值）的限制。
- **Default Centreon Graph Template（设置为默认图形模板）**：该选项允许 Centreon 设定该模板为系统默认的图形模板，便于那些之前未能够关联模板的服务项采用。

注意：值得注意的是，在上述选项中，我们始终没有见到有关性能图形坐标横轴的相关参数设定，这是因为 Centreon 的 RRD 图形模块在绘制图片时，始终以时间作为横轴坐标，因此无需设定。此外，在设定性能图形相关属性时，应尽可能使用同一套模板，这样可以使相关联的服务的性能图片保持视觉上的一致，有利于曲线的查看和问题的定位。

11.18.4 配置性能曲线相关属性

选择菜单 Views→Graphs，再单击左侧竖状菜单里的 Curves 链接，用户可以进入如图 11-82 所示的性能曲线模板列表界面。在该列表中，用户既可以通过单击 Add 链接的方式新增一项曲线模板，也可以基于已有的曲线模板复制出一套新的模板，还可以对既有的模板做删除操作，其管理方式与配置图形模板的方式一致。

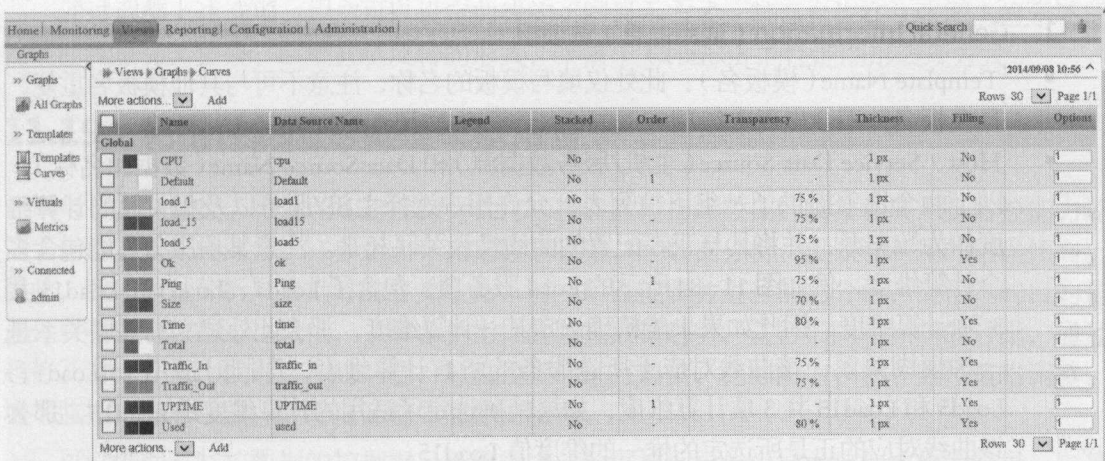


图 11-82 性能曲线模板

单击图 11-82 中的任意一项曲线模板名称，即可进入模板编辑页面，如图 11-83 所示。

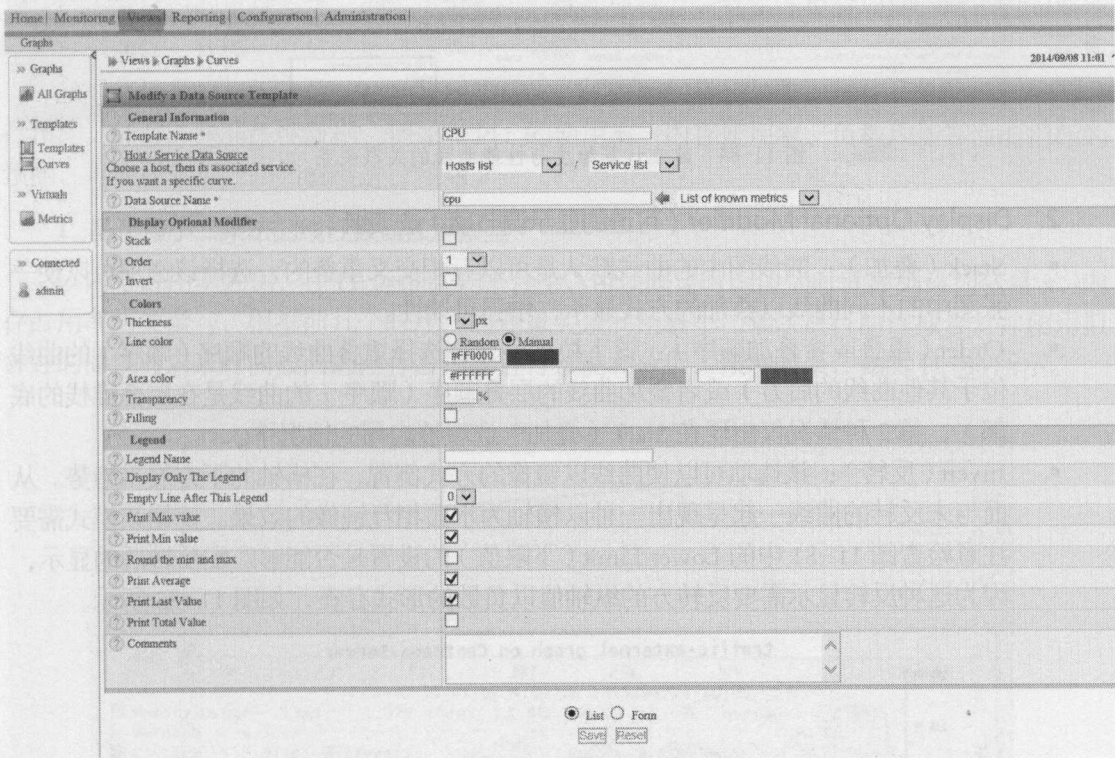


图 11-83 性能曲线模板编辑页面

如图 11-83 所示，性能曲线模板的配置页面由 4 类选项组构成。其中 General Information（通用信息）选项组用来配置曲线模板名和用来绘制曲线的性能数据源。Display Optional Modifier（可选的显示调节项）允许用户调整各类曲线相对于彼此之间的位置。Colors（颜色）选项用来设定各类曲线的颜色。Legend（图例）选项用来显示曲线的一些示例信息和注释信息。

1. General Information (通用信息) 选项组

- Template Name (模板名)：此处仅填写模板的名称，注意不可与其他模板名重复，且模板名并不会出现在绘制完成的性能图形中出现。
- Host / Service Data Source (主机/服务数据源) 和 Data Source Name (数据源名称)：以上两个选项提供了一组下拉列表，允许用户选择主机/服务以及相关联的计算维度作为数据源，并将其与正在配置的曲线模板关联起来。注意某项服务可能包含多个计算维度，例如图 11-84 所示的 Load 服务项，包含了 Load1、Load5 和 Load15 共 3 项计算维度。因此如果未能精确选定某一计算维度，那么此处建立的关联关系是一种模糊关系，该曲线对应于所选服务的所有计算维度，即同时适用于 Load1、Load5 和 Load15 共 3 项计算维度，如果精确选定了适用的计算维度是 Load15，那么该曲线对应的正是所选定的惟一的维度值 Load15。

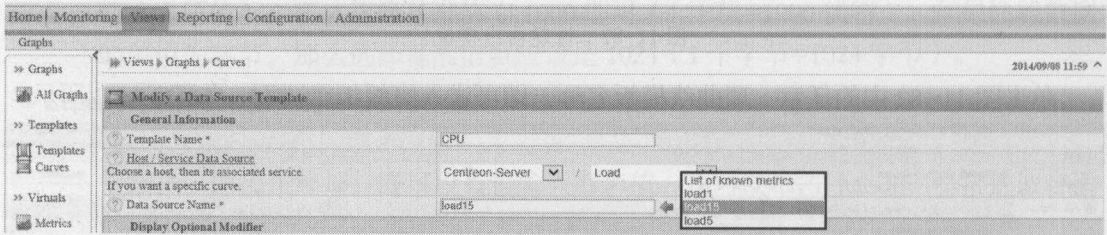


图 11-84 建立计算维度与性能曲线的关联关系

2. Display Optional Modifier (可选的显示调节项) 选项组

- Stack (叠加)：曲线图中的曲线默认是可以互相交叉重叠的，选择该选项可以使一张图中的不同曲线以叠加的方式显示，增强可读性。
- Order (重叠或者叠加顺序)：该下拉列表允许选择重叠曲线的顺序（顺序 1 的曲线位于其他曲线的后方）或者叠加曲线的层叠顺序（顺序 1 的曲线是在层叠堆栈的底部），第 2 种情况仅出现在 Stack（叠加）选项被启用的情况下。
- Invert (反转)：该选项可以使曲线以镜像的方式倒置，在横轴的下方展示趋势。从而与未反转的曲线一起呈现出一种以横轴为水面相互镜像的效果。使用该模式需要注意检查图 11-81 中的 Lower Limit（下限值）的设置是否能够匹配反转后的显示，因为这种反转显示需要反转方的纵轴值以负数的形式存在，如图 11-85 所示。

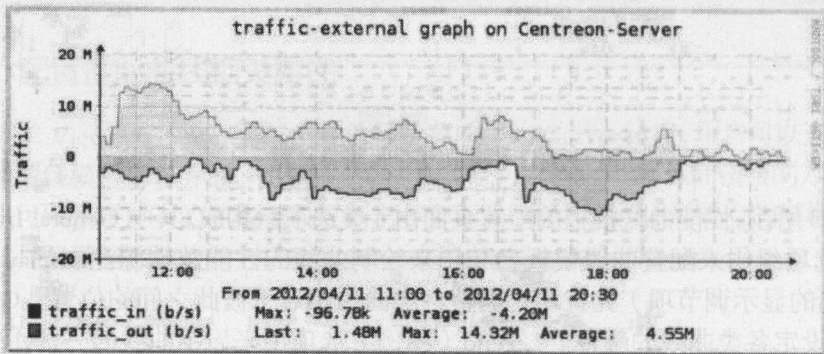


图 11-85 曲线的镜像反转

通过调整上述选项，用户可以设定曲线的不同显示方式，甚至允许在一张图上显示不同性能曲线呈交叉重叠、层叠甚至反转的形式，丰富了 Centreon 的性能图形表现方式。

11.19 利用性能图形实现早期预警

与日常的监报告警处理和监控指标优化一样，监控指标数据的收集和展示是监控能力评测工作的重要组成部分。许多企业存在重数据收集，轻数据分析、反馈和利用的错误倾向。因此，我们通过分析异常指标的各类数据，对 IT 监控服务进行能力分析，最根本的目的是通过对收集的数据进行量化，了解各类指标的监控数据趋势，进行横向和纵向的对比，通过趋势及对比分析，找到潜在的问题，对监控指标进行调整，比如阈值调整、监控策略调整、增加监控指标等手段，最终提升整体监控的质量。

可以借助对监控数据的挖掘提升预警能力。数据挖掘是信息技术自然演化的结果，它是从存放在数据库、数据仓库或其他信息库中的大量数据中挖掘有用知识的过程，其中数据预测是数据挖掘的重要目的之一，致力于为用户提供预警并辅助决策。主机服务器的性能一直是 IT 运维监控的重点工作，为了定位系统的性能瓶颈，探寻系统负载规律，对主机的性能数据进行监控、存储、分析并进行预测很有必要。

针对线性数据指标，例如磁盘空间使用率、用户数、进程数等，可使用线性回归模型进行预测。针对一些非线性数据，例如 CPU 的使用率、内存使用率、负载均衡等性能指标的预测，可使用非参数回归模型进行预测。

1. 基于最小二乘法的线性数据预测模型

以内存占用率为例，在 IT 运维监控平台中，可以固定时间段为时间序列设为变量 x ，内存占用率为变量 y ，根据前几个时段内的变量关系利用最小二乘法进行曲线拟合，并使用拟合好的曲线对未来时间段内的内存占用率进行预测和分析，预测效果，如图 11-86 所示。

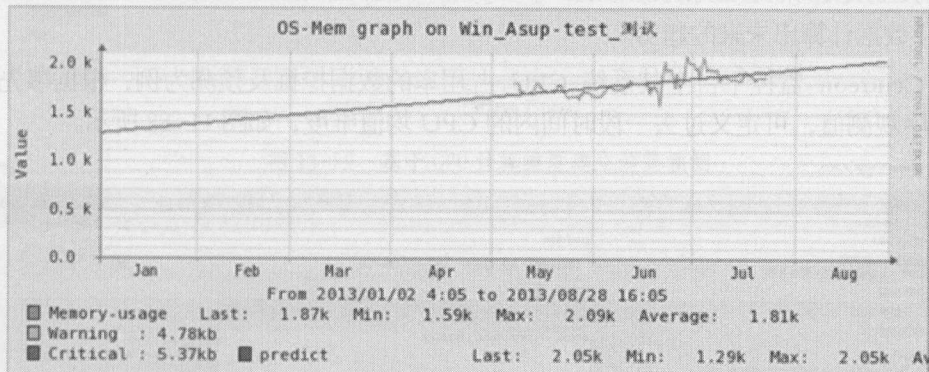


图 11-86 对监控数据做曲线拟合的样例

2. 基于滑动窗口平均值的 IT 系统性能预测模型

相对于线性预测而言，非线性的系统性能预测方法种类更多更复杂。企业长年累月的系统性能信息历史数据是非常丰富的，其中包含了大量的非线性系统负荷信息，更加有必要对其进行数据挖掘。发现时间序列数据库中蕴藏的相似性，有利于掌握数据变化规律和趋势，

为有效预测提供依据。

时间序列是指按时间顺序排列的观测值集合，如系统 CPU 性能信息。时间序列相似性搜索（又称为相似性）就是在时间序列数据库中发现与给定序列模式相似的序列或查找库中相似的序列对。例如，搜索今天与昨天、本月与上月负荷的相似性规律，从而帮助预测未来的负荷，如图 11-87 所示。

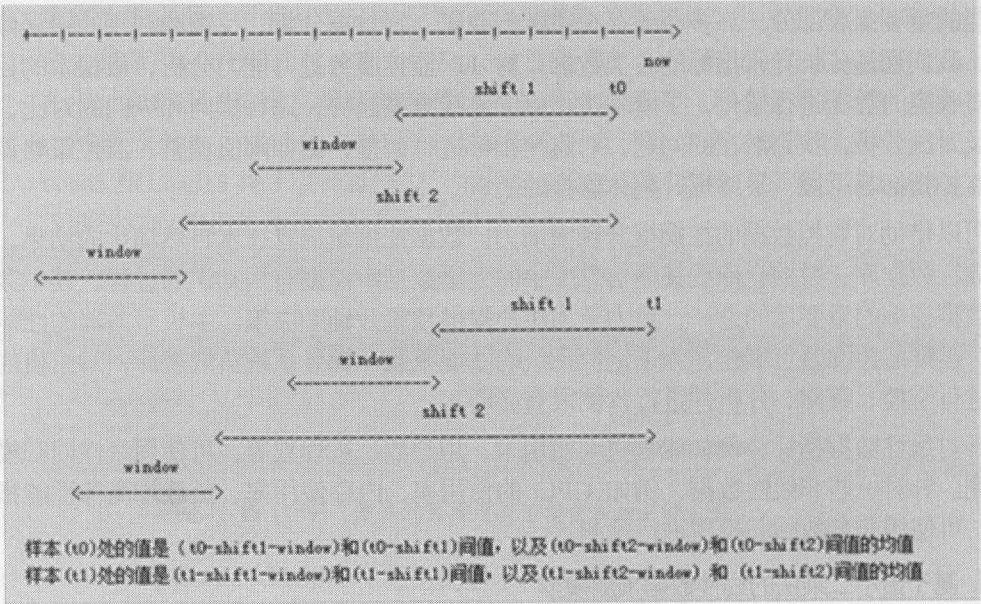


图 11-87 时间序列

可以根据 Centreon 平台中产生的 RRD 监控数据文件，结合 RRDTool 提供的图形函数（CDEF PREDICT 和 PREDICTSIGMA，可参考 <http://oss.oetiker.ch/rrdtool-trac/wiki/RRDtool14>），根据过往数据计算出未来的趋势。

以 Centreon 监控平台上某系统 CPU 占用率的数据挖掘及预测为例，根据事先搜集的 CPU 时序观测值，可定义过去一段时间内的 CPU 均值维度，如图 11-88 所示。

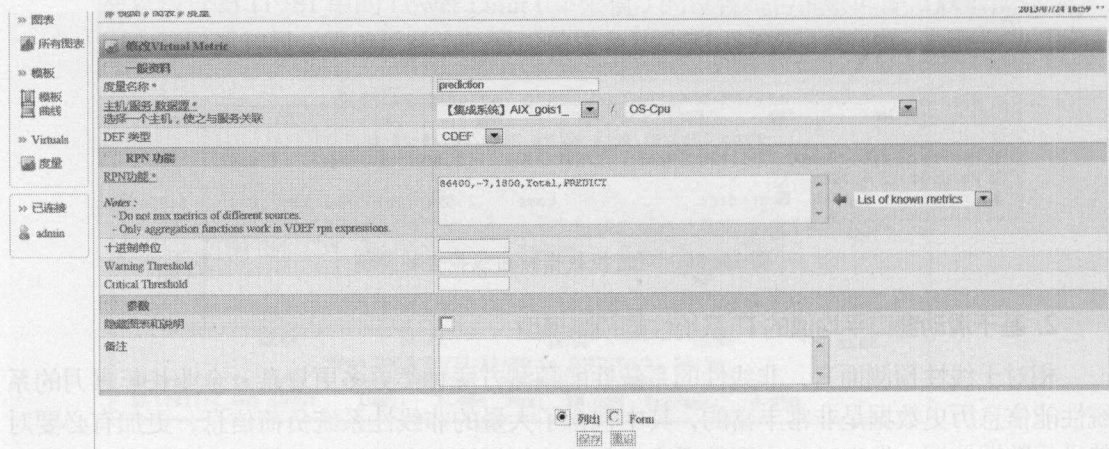


图 11-88 定义 CPU 均值计算维度

在上述定义中，CDEF 维度定义格式如下：

CDEF:predict_value = time, n, window, x, PREDICT

- time and n 代表从 t0 开始，向后滑动的时间。
- window 代表滑动窗口长度。
- x 代表待预测的样本数据变量名。

因此图 11-88 的定义可以解释为：

CDEF:predict_value = 86400, 7, 900, x, PREDICT

CPU 占用率数据预测将基于过去 7 天，间隔为 15 分钟的滑动窗口进行预测。

其中：

86400 = 60 seconds x 60 minutes x 24 hour

900 = 60 seconds x 15 minutes

而 PREDICTSIGMA 语法与 PREDICTD 的语法一致，它代表基于真实数据计算而得的数据平均值，所有预测数据都应该在 PREDICTSIGMA 值上下波动。

然后，在 CPU 变量上建立基于未来 3 天的趋势预测，如图 11-89 所示。

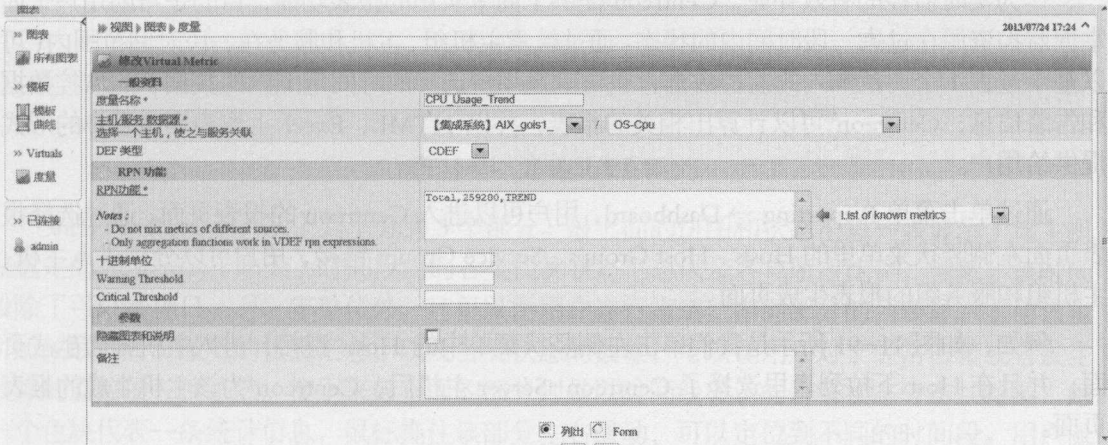


图 11-89 基于 CPU 计算维度建立趋势预测

图 11-90 所示是预测的效果：

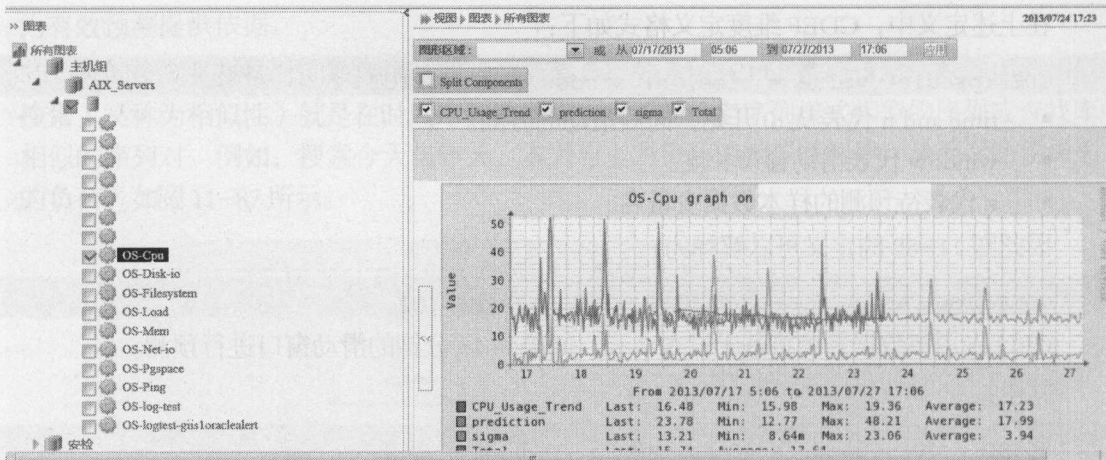


图 11-90 在 Centreon 平台建立监控项的趋势预测图形

11.20 报表

作为实时监控的有效补充，Centreon 提供了简单实用的报表功能，目的是帮助用户全面掌握监控资源在过去一段时间内的状态。通过检索主机组、主机和服务在一段时间范围内（可以基于过去 1 个月范围内的监控数据，也可以是用户选择的时间范围内的数据）的监控数据和告警信息，Centreon 可以计算出相关的报表，并以 HTML、Excel 工作表丰富多样的形式提供给用户。

通过单击菜单 Reporting → Dashboard，用户可以进入 Centreon 的报表页面。通过选择报表页面左侧竖状菜单里的 Hosts、Host Groups、Service Groups 链接，用户可以分别进入主机、主机组和服务组的报表生成页面。

例如，如图 11-91 所示是我们单击左侧竖状菜单中的 Hosts 链接，进入主机报表生成页面，并且在 Host 下拉列表里选择了 Centreon-Server 主机后，Centreon 为该主机生成的报表页面。



图 11-91 主机报表页面

如图 11-91 所示，从上到下分为 3 个部分。最上面的饼图和状态汇总列表显示了该主机在选定的 Yesterday（昨天）时间段内的各类检测状态汇总以及各自所占比例，当然，该时间段除了可以选择日、月、年单位外，还可以由用户自行指定。中间部分的列表显示了与该主机所关联的所有服务的状态汇总，以及各类状态的数量，可以让用户对于经常告警或者发生故障的监控项一目了然。接下来最下面的部分则是以时间线显示的告警状态分布甘特图，每一个色块代表一条统计信息，鼠标拖住该部分左右滑动，可以定位到不同的时间段。用户鼠标选择色块，可以了解该色块所代表的详细统计信息。



第 12 章

Centreon 的管理和优化

本章聚焦 Centreon 管理和优化的相关操作与技巧。



12.1 Centreon 的管理菜单

对于 Centreon 的管理可以通过选择 Administration 菜单来实现，如图 12-1 所示。

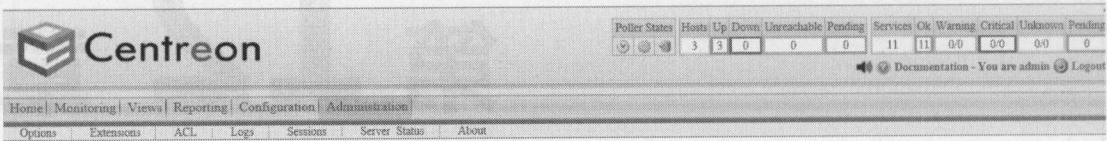


图 12-1 Centreon 的管理菜单

如图 12-1 所示，整个 Centreon 的 Administrations（管理）菜单分为 7 个子菜单：

- Options（选项）：该子菜单提供了 Centreon 监控平台的绝大多数重要参数，很多重要的全局参数都可以通过该菜单管理。
- Extensions（扩展）：该子菜单提供了 Centreon 监控平台的插件管理功能。Centreon 具备很强的扩展功能，可以通过多种类型的插件实现功能的提升。该子菜单即提供了插件的管理界面，可进行插件的列表查看、安装、卸载等功能。
- ACL（访问控制）：该子菜单提供了 Centreon 的权限管理和访问控制列表管理功能。
- Logs（日志）：该子菜单所提供的日志记录了所有用户通过 Centreon 所做的所有管理操作，提供了 Centreon 监控平台的审计功能。
- Sessions（会话）：该子菜单提供了目前正在访问 Centreon 平台的所有用户会话列表，并且提供了相关选项以便于管理人员随时终止某一个用户的会话。
- Server Status（服务器状态）：提供了 Centreon 监控平台的统计信息，以及监控平台所在中央监控服务器的相关系统信息，向管理人员提供了 Centreon 监控平台的软硬件信息总览。
- About（关于）：提供了 Centreon 的开发者信息和版权信息等。

12.2 通用选项

通过选择菜单 Administration → Options，可进入 Centreon 最为重要的通用选项管理界面。这些管理选项以左侧竖状菜单里的项和子项的形式进行组织，如图 12-2 所示。

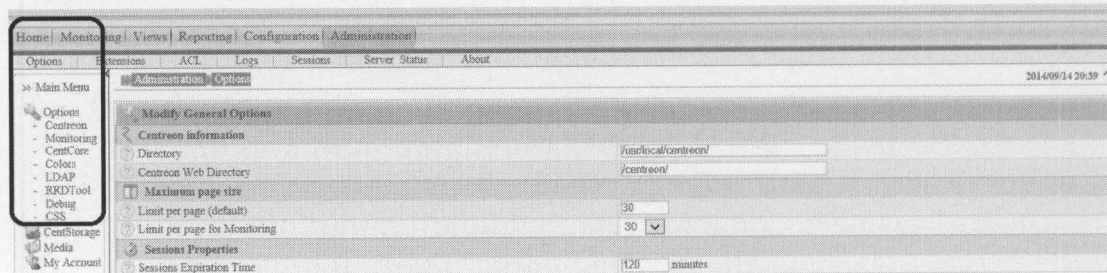


图 12-2 Centreon 的通用选项管理菜单

12.2.1 Centreon 的通用选项界面

接下来的 Web 界面是各项参数的管理配置页面，首先是 Centreon 的通用选项界面，如图 12-3 所示。

The screenshot displays the 'Modify General Options' page in the Centreon web interface. The top navigation bar includes links for Home, Monitoring, Views, Reporting, Configuration, and Administration. Below this, a secondary bar shows Options, Extensions, ACL, Logs, Sessions, Server Status, and About. The left sidebar contains a 'Main Menu' with links to Options, Monitoring, CentCore, Colors, LDAP, RRDTool, Deflog, CSS, CentStorage, Media, and My Account. The main content area is titled 'Administration > Options' and shows the date '2014/09/14 20:44'. The configuration is organized into several sections: 'Centreon information' (Directory, Centreon Web Directory), 'Maximum page size' (Limit per page (default), Limit per page for Monitoring), 'Sessions Properties' (Sessions Expiration Time), 'Refresh Properties' (Refresh Interval, Refresh Interval for statistics, Refresh Interval for monitoring, First Refresh delay for statistics, First Refresh delay for monitoring), 'Display Options' (Display Template), 'Display properties' (Sort by, Order sort), 'Problem display properties' (Sort problems by, Order sort problems), 'Authentication properties' (Enable Autologin, Display Autologin shortcut, Enable SSO authentication, SSO mode, SSO trusted client addresses, SSO login header), 'Time Zone' (Enable Timezone management, Default host timezone), 'Configuration UI behavior' (Enable strict mode for host parentship management), and 'Support Information' (Centreon Support Email). At the bottom, there are 'Save' and 'Reset' buttons.

图 12-3 Centreon 的通用选项管理界面

- **Centreon information (Centreon 信息)**：该选项组用以配置 Centreon 在 Web 应用服务器中相关路径信息。其中 Directory (目录) 选项填写 Centreon 在中央监控服务器上的实际部署目录，而 Centreon Web Directory (Centreon Web 路径) 选项用来设置通过浏览器访问 Centreon 监控平台时使用的 Web 根路径。
- **Maximum page size (最大页面数)**：该选项组中的 Limit per page (default) (每页限制对象数量 (默认)) 选项用来设置在主机、服务、命令等列表界面中，每页默认显示的对象数量。如图 12-3 所示该项默认为 30，意味着在主机列表页面中，每页出现的主机有 30 项，如果 Centreon 监控平台中总共监控的主机不超过 30 台，那么在 1 页中即可显示全部主机；如果超过 30 台，那么就需要分页显示。Limit per page for Monitoring (限制监控页面中的列表对象数量) 选项用以设置在 Monitoring→Services 或者 Monitoring→Hosts 菜单所示的实时监控页面中能够显示的最大对象数量。因为这些实时监控页面为后台不断刷新以能够显示最新状态的页面，如果显示对象数量过多，会对 Centreon 中央监控服务器的性能造成影响，因此有必要根据终

端浏览器刷新的速度限制其只能够在 1 页内显示一定数量的对象。

- Sessions Properties（会话属性）：其中的 Sessions Expiration Time（会话过期时间）选项设定了 Centreon 监控平台中用户会话的过期时间，如果用户已经登录 Centreon，而在此时间段之内没有任何动作，那么过了该时间段之后，Centreon 会移除该用户会话，用户若想继续访问 Centreon，需要重新登录。
- Refresh Properties（刷新属性）：该选项组可以用来设置页面自动刷新的相关属性，例如实时监控的页面，默认 60 秒刷新一次，以及 Centreon Web 用户界面顶部的相关计数器，默认 15 秒刷新一次。
- Display Options（显示选项）：其中的 Display Template（显示模板）下拉列表允许用户为 Centreon 的 Web 用户界面选择预先定义好的主题，从而设置不同的显示风格，如图 11-79 所示。目前系统仅提供了 Centreon-2 主题。
- Display properties（显示属性）：设置实时监控页面中列表显示的排序方式，默认以 Host（主机）作为排序项，且为顺序排序。而 Sort problems by（问题对象排序）下拉列表则提供了 Duration（故障持续时间）、Service（服务）、Status（状态）、Output（输出）等项作为排序项。
- Authentification properties（认证属性）：该选项组中的 Enable Autologin（自动登录）选项允许用户不需要输入用户名和密码，而是通过一个超链接直接登录 Centreon 监控平台。在监控实践中，一般将 Centreon 监控平台中权限较低的只读用户名，以及加密过的密码配置在某个监控图标的超链接中，这样可以通过单击该图标就可以自动登录 Centreon 监控平台，并定位到相应的主机或者服务上（参考 13.9 小节）。

如图 12-4 所示，单击代表某台主机的图标，该图标的超链接就可以直接显示到 Centreon 监控平台的相关链接。

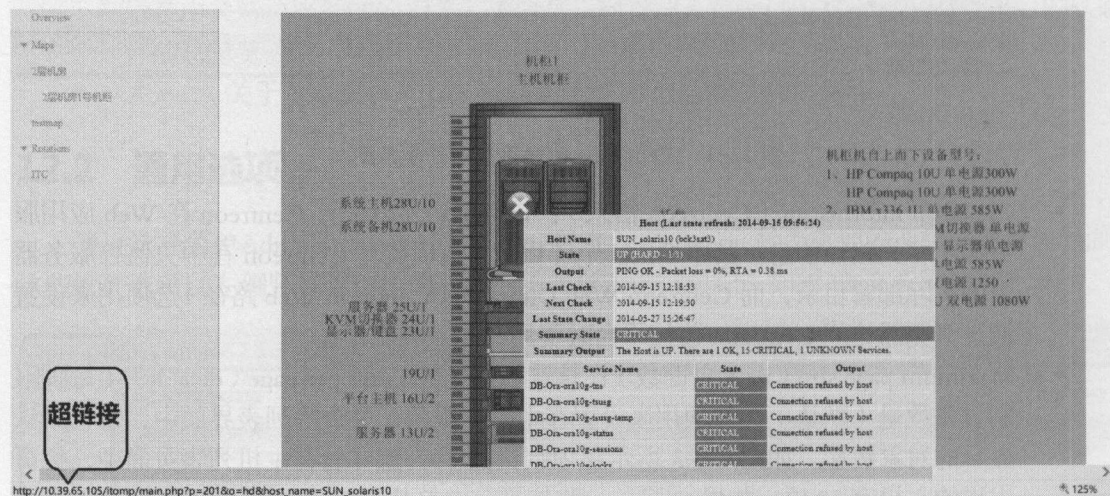


图 12-4 图标到 Centreon 监控平台的超链接

注意：自动登录选项只能适用于 Centreon 监控平台中权限较低的用户，最好是只读用户。为系统安全起见，绝对不能将具备管理功能的用户，尤其是管理员用户配置为自动登录。

- Time Zone (时区)：该选项组用来管理时区并设置默认的时区值。如果启用时区管理功能，那么 Centreon 监控平台中的用户将会具备时区属性，并且可以设置各自所在时区偏移值，从 -12 到 12，其中 0 代表格林尼治时间。

12.2.2 Centreon 的监控选项界面

选择菜单 Administration→Options，在左侧竖状菜单单击 Options 链接，如图 12-5 所示。

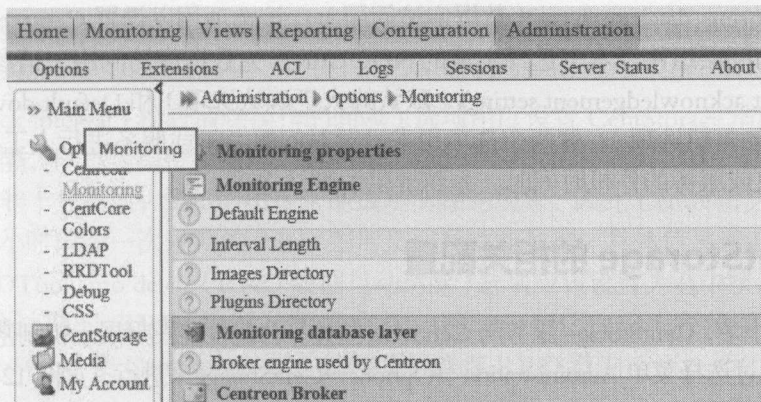


图 12-5 Centreon 的监控选项链接

单击 Options 链接之后，可进入如图 12-6 所示的 Centreon 监控相关选项的管理界面。

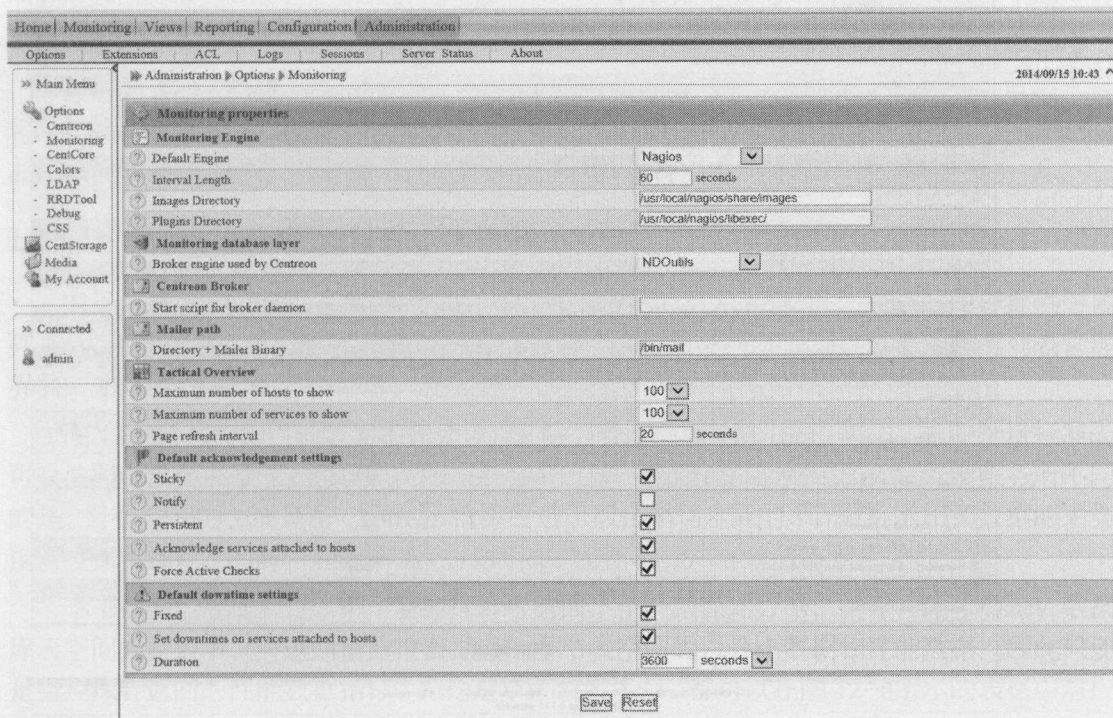


图 12-6 Centreon 的监控选项管理界面

- Monitoring Engine（监控引擎）：该选项用来为 Centreon 监控平台设置默认的调度引擎，即 Nagios。其中的 Images Directory（图像目录）选项用以设置 Nagios 的图标文件存储路径。与 Centreon 一样，Nagios 也提供了可供浏览器访问的 Web 界面。在 Nagios 的 Web 用户界面中，需要相应的图标显示，该路径指定的目录即用来存储这些图标。Plugins Directory（插件目录）选项用来设定 Centreon 监控插件所存放的服务器上的目录，这些插件在配置 Centreon 的检测命令时需要用到，可以参考 11.5 小节。
- Monitoring database layer（数据库访问层）：该选项用以设置 Centreon 所默认的数据库访问层组件，本书选择主流的 NDOUtils 作为数据库访问层，请参考 9.3 小节。
- Default acknowledgement settings（默认的人工确认选项）和 Default downtime settings（默认的停机选项）：这两个选项用来设置对问题主机或服务进行人工确认，以及设置主机或服务的停机时间时，系统提供的默认选项。

12.3 CentStorage 的相关配置

参考 7.6.2 小节，CentStorage 服务与 Centreon 监控平台的性能图形展示机制密切相关，对其相关配置可以通过选择菜单 Administration→Options→CentStorage来进行，如图 12-7 所示。

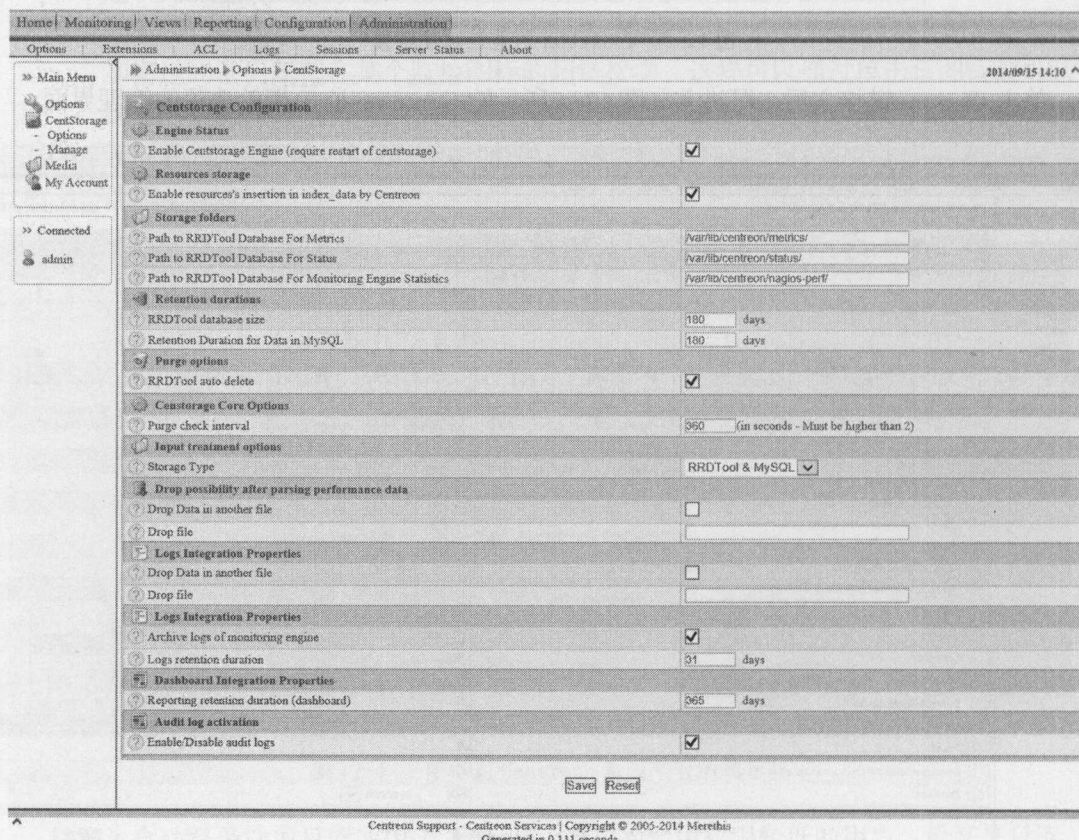


图 12-7 CentStorage 的相关配置

绝大多数情况下,在 Centreon 监控平台中,对于性能图形数据的处理是通过 CentStorage 组件执行的,但是也可以通过例如 Centreon Broker 等其他专有组件来完成这项任务。一旦通过其他组件,而非 CentStorage 来执行性能图形数据处理任务,那么就需要设置 Enable Centstorage Engine(require restart of centstorage) (启用 CentStorage 引擎(需要重启 centStorage 服务)) 选项为未选中状态。

由于数据库空间、文件系统存储空间的有限性, Centreon 中可以对性能图形的存放周期进行配置,在选项组 Retention durations (保留时间) 中,可以分别对 CentStorage 组件所生成 RRD 文件数据库的存放天数,以及后台 MySQL 数据库中性能数据存放天数进行设置,默认均为 180 天。

Purge check interval (检查清除操作的间隔) 选项用以设置 CentStorage 清除过期的 RRD 相关数据的间隔,以秒为单位。需要注意的是,在一个较大的部署环境中, CentStorage 执行清除操作的并非十分有效,偶尔会出现未能成功清除的现象,因此 Centreon 还提供了一项定时任务,每 2 小时清除一次过期的 RRD 图形文件及性能数据,具体信息请参考 7.6.3 小节。接下来的 RRDTool auto delete (RRD 数据自动清除) 选项应该设置为选中状态,使 Centreon 能够自动清除过期的 RRD 数据及相应文件。而 Storage Type (存储类型) 应设置为 RRDTool&MySQL, 使 CentStorage 组件能够正确操作 RRD 数据。

Drop Data in another file (允许 RRD 数据导出为文件) 选项可以允许经处理的性能数据导出为文件,而导出文件路径可以通过 Drop file (导出文件路径) 选项设置。该功能在引入外部数据处理机制时特别有用,可以将 CentStorage 生成的 RRD 数据导出为文件,并传输另一外部数据处理组件来进行特别处理,从而扩展了 Centreon 监控平台的接口,实现了采集数据的输出。

最后的 Archive logs of monitoring engine (归档 Nagios 日志) 选项允许 Centreon 监控平台将 Nagios 运行日志归档保存,默认路径是 /usr/local/nagios/var/archives。而 Logs retention duration (Nagios 日志保存时间) 选项则设置了这些运行日志的过期时间,默认为 31 天。

12.3.1 性能数据的配置管理

对于这些由 CentStorage 组件处理完毕的性能数据的展示和管理, Centreon 同样提供了系列工具,可以通过菜单 Administration → Options → CentStorage → Manage 进入性能数据列表管理界面,如图 12-8 所示。

图 12-8 所列出的性能数据指标列表,又称为计量标准 (Metrics) 列表。这些计量标准既可以按照主机名来过滤,还可以按照服务名来过滤。与显式声明并定义主机以及服务所不同的是,计量标准的生成是由 Centreon 监控平台于后台悄悄生成的,计量标准相关数据的采集由监控探针执行,而解析和入库由 CentStorage 或者 Centreon Broker 组件来完成。

本书前面已经提到过,单个监控探针可返回不止一项性能数据,例如,针对 Oracle 数据库表空间的监控项,可返回被监控数据库的多个表空间的百分比数据和性能数据。因此只要是监控探针返回的性能数据指标,无论是单个或者多个,都可以在图 12-8 所示的表格中对应一项计量标准,且这些计量标准的单位显示在括号中。

Host	Service	Metrics	Rebuild Waiting	Delete	Hidden	Locked	Storage Type
Centreon-Server	Disk-/	/ (MB)	No	No	No	No	RRDTool & MySQL
	Disk-opt	opt (MB)	No	No	No	No	RRDTool & MySQL
	Disk-var	var (MB)	No	No	No	No	RRDTool & MySQL
	Load	load1 - load15 - load5	No	No	No	No	RRDTool & MySQL
	Ping	pi (%) - rta (ms) - rtmux (ms) - rtmin (ms)	No	No	No	No	RRDTool & MySQL
	Swap	swap (MB)	No	No	No	No	RRDTool & MySQL
	Swap_1	swap (MB)	No	No	No	No	RRDTool & MySQL
	Swap_1_1	swap (MB)	No	No	No	No	RRDTool & MySQL
	Swap_1_1_1	swap (MB)	No	No	No	No	RRDTool & MySQL
localhostnag	Swap	swap (MB)	No	No	No	No	RRDTool & MySQL

图 12-8 性能数据指标管理界面

图 12-8 所示的表格提供了正在运行的 Centreon 监控项的实时状态信息，如下所示：

- Rebuild Waiting（重建等待）：该列显示了 Centreon 的图形组件是否正在基于后台数据库中的监控性能数据而构建性能图形。
- Hidden（隐藏）：该列显示了该监控性能指标是否为除了系统管理员之外其他用户不可见。
- Locked（锁定）：该列显示了监控性能指标是否不再由后台 RRD 数据库所提供。
- Storage Type（存储类型）：显示了监控性能指标的存储位置，一般是 RRD 数据文件和后台 MySQL 数据库。

单击上述表格中的服务名，还可以进入该服务的监控性能指标的详细配置页面，如图 12-9 所示。

Metric	Unit	Warning	Critical	Min	Max	Data source type	Hidden	Locked
opt	MB	927	917	0	0	GAUGE	No	No

图 12-9 监控指标的详细配置页面

在图 12-9 中的配置页面中，可以对服务监控指标阈值、告警级别、数据源类型（基于度量、或是基于计数器，后续有介绍）、是否隐藏以及是否锁定等状态进行进一步调整。

12.3.2 度量和计量

图 12-9 中提到的，每项监控计量标准都有自己所属的数据源类型，即度量方式（Gauge），或者是计量方式（Counter）。数据源类型也是监控探针所返回的监控数据的形式，通常反映了监控数据是以何种形式被监控项所呈现出来。RRDTool 组件需要基于这些监控数据及其类型来生成 Centreon 中所展现的性能图形。

1. 度量方式 (Gauge)

度量方式是将监控探针返回的监控数据与预先定义好的阈值进行比较，该方式称为度量。度量方式是最为普遍使用的监控标准，适用于文件系统利用率监控、CPU 利用率监控，以及 Ping 值延迟监控等多种标准参数的监控。

2. 计量方式 (Counter)

计量方式是指监控探针返回的监控数据是一个逐渐累加的数字，呈不断上升的计数方式。计数方式主要用于衡量该计量标准逐渐聚集的程度。例如，通过 SNMP 方式获取的网卡传输的性能数据主要有输入字节数和输出字节数两种，而 SNMP 协议仅仅计算该网卡的瞬时输入或者输出流量，为了计算该网卡的传输速率，我们必须要以计量方式汇总流经该网卡的所有字节数（逐渐累加不断增长的字节数）来计算该网卡的实际传输速率。

12.3.3 监控性能指标的相关操作

在图 12-8 的监控指标详细配置页面中，通过单击 More actions 下拉列表，可进入计量标准 (metric) 的管理菜单，如图 12-10 所示。

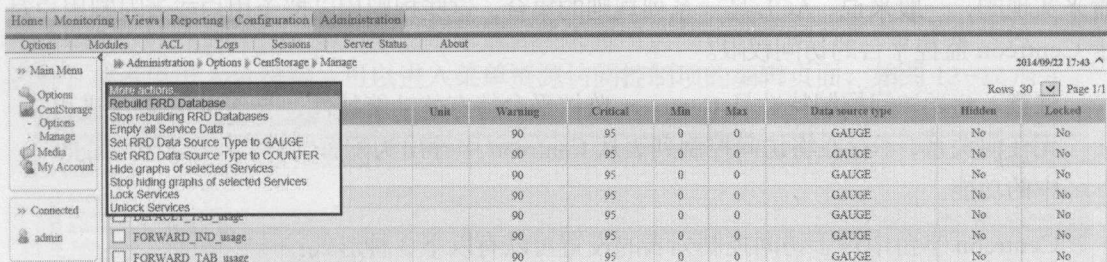


图 12-10 监控指标的管理菜单

- **Rebuild RRD Database (重建 RRD 数据库)**
在出现网络延迟或者数据包丢失的情况下，检测探针返回的监控项性能数据就会存在缺失，继而由 RRDTOOL 组件处理并生成的性能数据图形相对于连续的时间就会出现“空洞”的现象。在缺失数据量不大的情况下，可以使用“重建 RRD 数据库”选项来填补性能数据图形，使其尽量呈现连续的图形。如果缺失的性能数据过多，或者时间断续过大，那么该项无效。一旦调用该选项，Centreon 就会删除文件系统中原有与该监控计量项相关联的 RRD 文件，重新抽取 MySQL 数据库中的监控数据，并调用 RRDTOOL 组件来生成该监控项的性能图形。
- **Empty all Service Data (清空所有服务性能数据)**
在遇到服务项性能数据返回错误，或者性能数据的单位发生改变的情况下，使用该选项可将后台 MySQL 数据库中的性能数据以及据此生成的性能图形文件全部清空，以便后续接纳正确的性能数据，生成正确的性能图形文件。
- **GAUGE/COUNTER (更改数据来源类型)**
Centreon 在根据监控项返回的性能数据生成计量标准时默认采用的数据来源类型为 GAUGE 型，即度量方式。如果度量方式需要修改，只需选中该计量标准，并更改其数据来源类型为 COUNTER (计量) 即可。注意修改后，需要执行 Empty all Service

Data(清空所有服务性能数据)操作, 并接着执行 Rebuild RRD Database(重建 RRD 数据库) 操作以重新生成该监控项的性能图形。

- Hide graphs of selected Services (隐藏选中服务的性能图形)

该操作可以是临时的行为, 使管理人员能够隐藏某些正常的计量标准的图形, 将注意力专注于解决存在问题的性能图形上。也可以隐藏某些异常的计量标准的性能图形, 以减少不必要的告警数量。

- Lock Services (锁定服务)

该操作会暂时冻结服务及其计量标准的性能图形展示, 但是并不会影响 Centreon 平台在后台生成性能图形, 且被冻结服务及计量标准的性能数据和性能图形都不会丢失。

12.4 访问控制列表 (ACL)

正如 1.4 小节中对于 Centreon 的介绍中所提到的, Centreon 能够基于自身的访问控制列表 (Access Control List, ACL) 实现精细化的访问管理, 通过设置安全策略来确保用户只能看到被授权访问的资源, 从而实现自定义的访问控制。访问控制是通过对于 ACL 的管理和配置来实现的, 一般来说, ACL 是一系列规则的集合, 这些规则用以授予用户或者限制用户对于 Centreon 监控平台的访问权限。

注意：此类访问控制特性是 Centreon 监控平台特有的功能特性, 与 Nagios 调度进程无任何关系, 不存在将访问控制列表从 Centreon 中导出为配置文件, 并导入到 Nagios 中的功能。

Centreon 平台中对于访问控制列表的设置可以有以下 3 种形式:

- 授予或者限制用户对于 Centreon 的 Web 界面菜单的访问。这是 Centreon 访问控制列表最易实现的功能。通过授予或者限制用户对于 Centreon 的 Web 用户界面菜单、子菜单、类别以及界面左侧竖状菜单中相关选项的访问权限, 可以授予或者限制用户对于 Centreon 中某 (些) 项功能的访问, 从而实现访问控制的目的。
- 授予或者限制用户对于 Centreon 平台中的监控资源, 即主机或者服务等资源的访问。此类设定可以管理用户对于部署在 Centreon 监控平台中的主机组、主机、服务组、服务、元服务等监控资源的访问权限。
- 授予或者限制用户对 Centreon 中某些功能的访问。设定用户能够执行某些 Centreon 用户界面中的命令或者按钮, 以及限制用户执行某些菜单选项, 可以让具备不同角色的用户具备不同的管理权限, 以实现对于 Centreon 监控平台的精细化、安全化的管理。

在 Centreon 中, 只有具备管理员权限的用户才能访问 Centreon 平台 Web 用户界面中的所有功能菜单、查看所有主机组、主机、服务组、服务等监控资源, 执行各类 Centreon 命令。如果非管理员用户想要访问 Centreon 的 Web 用户界面, 查看某些资源并执行某些操作, 必须经管理员配置相应的访问控制权限, 才能具备相应的功能。

一般地, 对于访问控制权限的设置并非立竿见影, 需要经过约 2 分钟的调度之后才能生效。对于已经登录 Centreon 平台的用户来说, 对其访问权限的设置无法实时生效, 该用户必

须退出并重新登录 Centreon 后，其访问权限才能生效。

12.4.1 访问控制列表的配置与管理

接下来我们了解一下如何配置访问控制列表，以实现对于 Centreon 菜单资源的访问控制。

1. 菜单资源的访问控制

选择系统菜单Administration→ACL，并单击左侧竖状菜单中的 Menus Access 子菜单，进入 Centreon 菜单资源的访问控制页面，如图 12-11 所示。

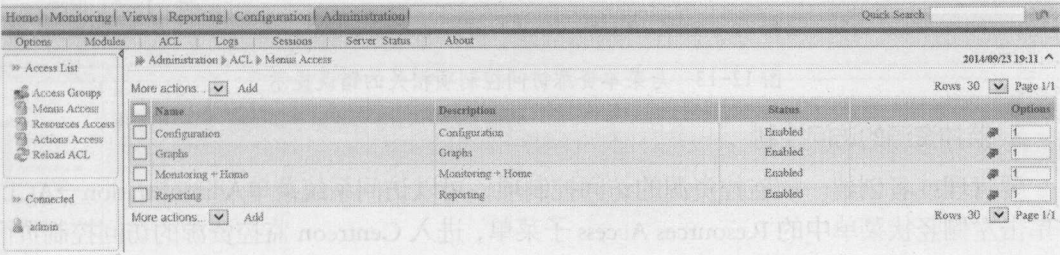


图 12-11 Centreon 的菜单访问控制项列表

图 12-11 中列举出了系统中已经配置完毕的菜单资源访问控制列表项。单击任何一项的名称，或者单击 Add 链接，可以进入菜单资源访问控制项的编辑页面，如图 12-12 所示。

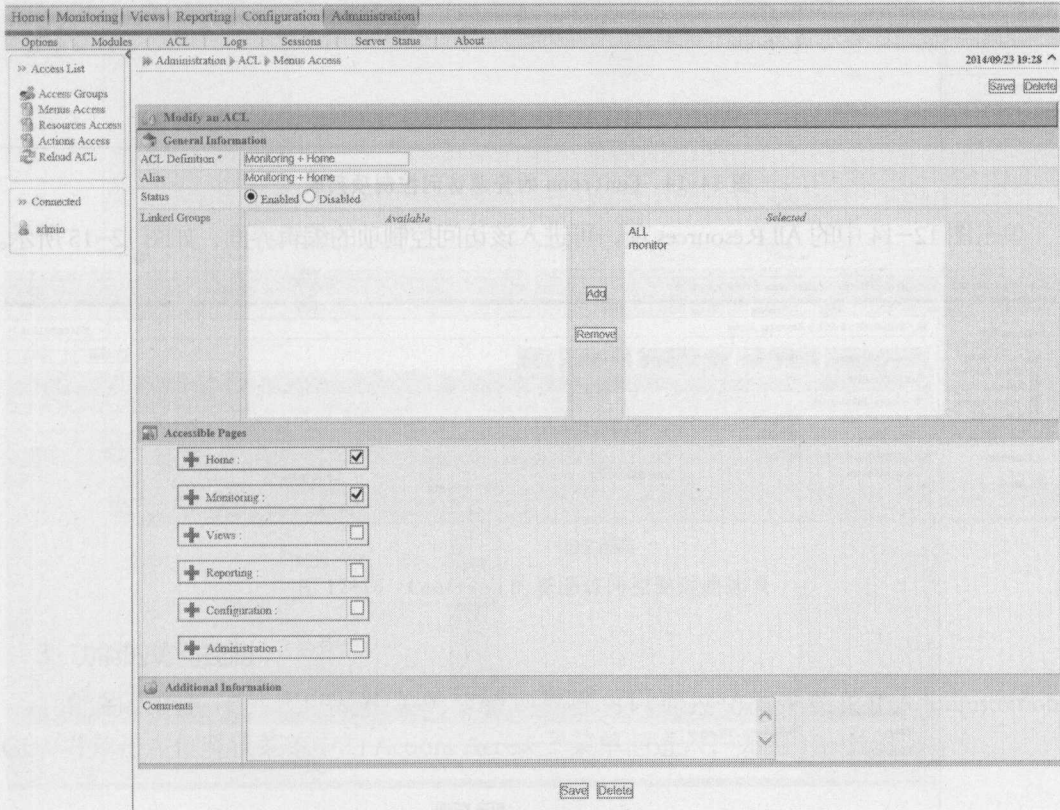


图 12-12 Centreon 的菜单访问控制项的编辑页面

图 12-12 中的界面显示了菜单资源及相关访问组（Access Group）的关联关系配置。在

Linked Group 选项卡中，我们选择了 monitor 和 ALL 共两项访问组，在 Accessible Pages 树状选项卡中，我们复选了 Home 和 Monitoring 两项菜单。选择完毕后，在 ACL Definition 中填写系统唯一的访问控制项名称，单击 Save 按钮，即定义了一项新的菜单资源访问控制项，意味着位于 ALL 和 monitor 两项访问组中的任何一个 Centreon 平台用户，都具备访问 Centreon Web 页面中的 Home 和 Monitoring 共两个菜单项。而单击其他菜单项时，将会显示 You are not allowed to reach this page（不允许访问该页面）的错误，如图 12-13 所示。

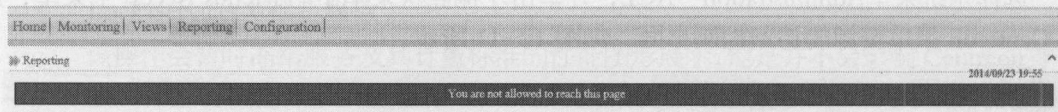


图 12-13 与菜单资源访问控制项相关的错误提示

2. 监控资源的访问控制

要管理或者创建一项监控资源的访问控制项，可以访问系统菜单 Administration→ACL，并单击左侧竖状菜单中的 Resources Access 子菜单，进入 Centreon 监控资源的访问控制页面，如图 12-14 所示，系统默认提供了 All Resources 的资源访问控制项。

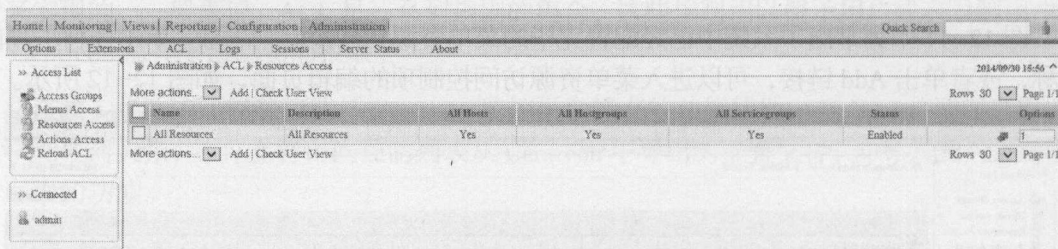


图 12-14 Centreon 的资源访问控制项列表

单击图 12-14 中的 All Resources 项，可进入该访问控制项的编辑界面，如图 12-15 所示。

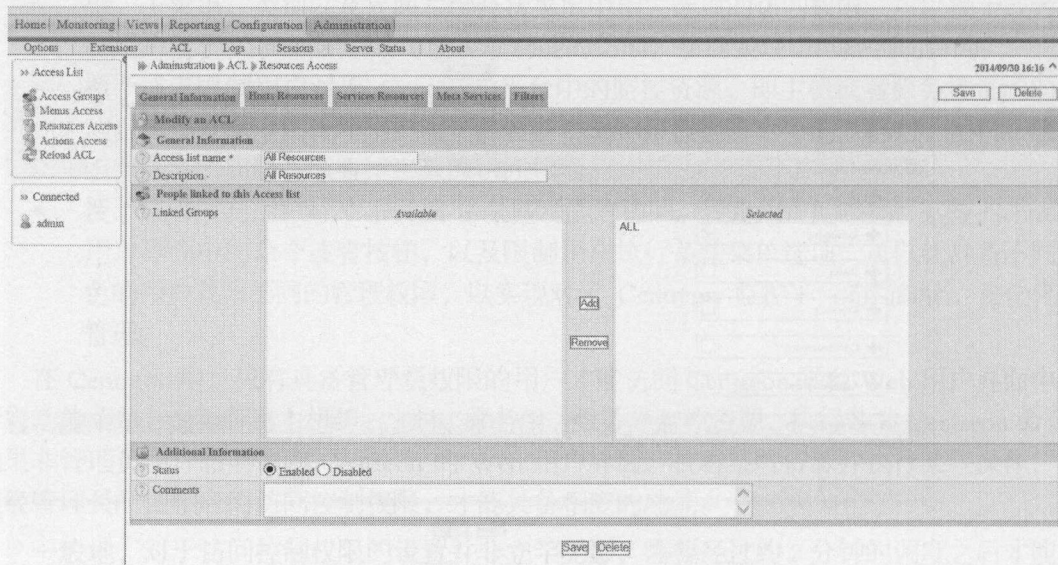


图 12-15 Centreon 的资源访问控制项编辑界面

上述编辑页面中，General Information 选项卡用来设定该访问控制项的名称，以及相关用户的访问组，其他选项卡用来关联主机组、服务组、元服务以及过滤选项。

在最后的 Filter 选项卡中，管理员可以为之前已经关联过的主机、服务等资源设置过滤项，以更精确地实现进一步的访问控制。在一个分布式的监控架构中，位于不同服务器上的 Nagios 调度进程负责采集不同类型的监控资源信息，例如各自负责采集数据库信息和网络信息，那么通过设置 Poller Filter 过滤项，即可控制不同用户对于不同类型监控资源的访问，如图 12-16 所示。

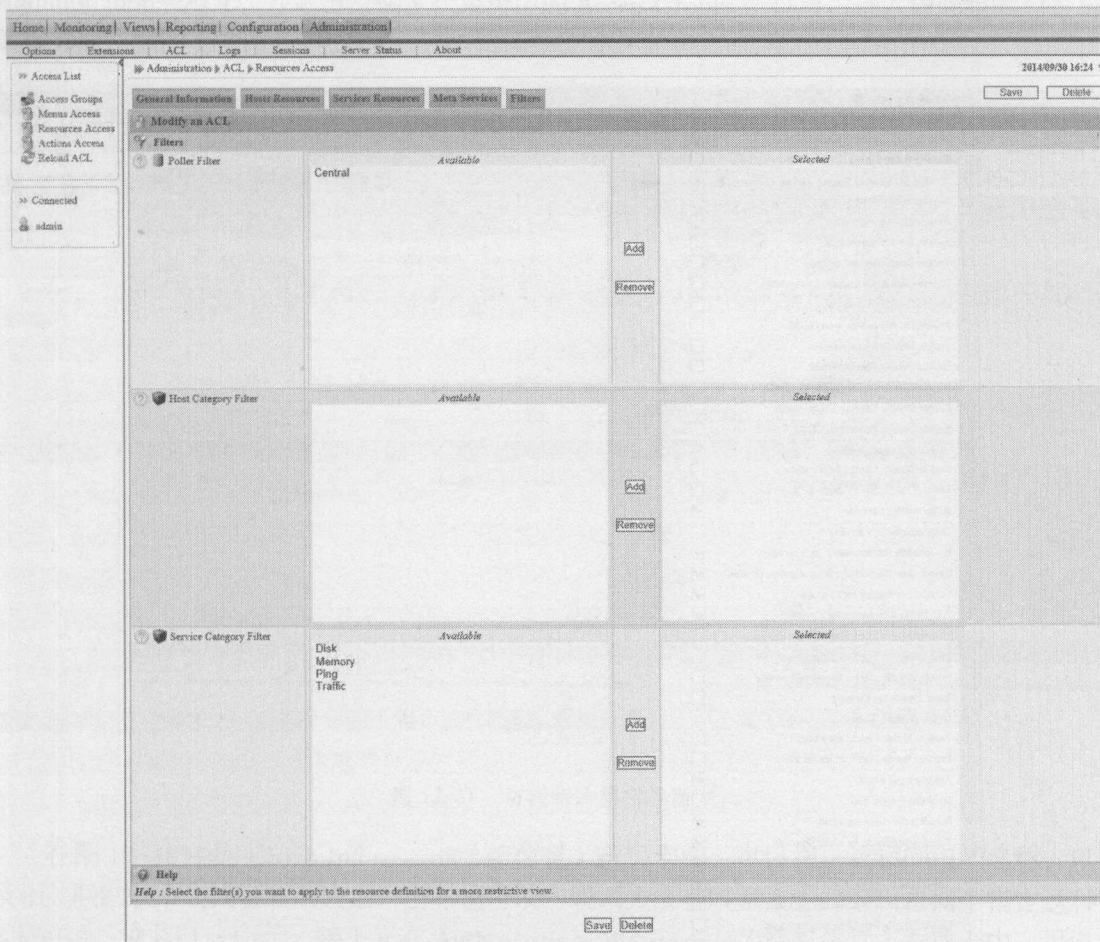


图 12-16 Centreon 的资源访问控制过滤选项

3. 功能的访问控制

功能的访问控制列表以及相关的编辑项界面可以通过访问系统菜单 Administration → ACL，并单击左侧竖状菜单中的 Actions Access 子菜单而进入，如图 12-17 所示。

Home | Monitoring | Views | Reporting | Configuration | Administration

Options | Extensions | ACL | Logs | Sessions | Server Status | About

» Access List

» Administration » ACL » Actions Access

2014/09/30 16:51

Save | Reset

» Access Groups

» Menu Access

» Resources Access

» Actions Access

» Reload ACL

» Connected

admin

» Modify an Action

» General Information

Action Name * Single User

Description * Single User

» Relations

Linked Groups * ALL

ADD

Delete

» Global Functionalities Access

Display Top Counter ☒

Display Top Counter pollers statistics ☒

Display Poller Listing ☐

» Global Monitoring Engine Actions (External Process Commands)

Shutdown Monitoring Engine ☐

Restart Monitoring Engine ☐

Enable/Disable notifications ☐

Enable/Disable service checks ☐

Enable/Disable passive service checks ☐

Enable/Disable host checks ☐

Enable/Disable passive host checks ☐

Enable/Disable Event Handlers ☐

Enable/Disable Flap Detection ☐

Enable/Disable Obsessive service checks ☐

Enable/Disable Obsessive host checks ☐

Enable/Disable Performance Data ☐

» Services Actions Access

Enable/Disable Checks for a service ☐

Enable/Disable Notifications for a service ☐

Acknowledge a service ☒

Disacknowledge a service ☐

Re-schedule the next check for a service ☒

Re-schedule the next check for a service (Forced) ☒

Schedule downtime for a service ☒

Add/Delete a comment for a service ☒

Enable/Disable Event Handler for a service ☐

Enable/Disable Flap Detection of a service ☐

Enable/Disable passive checks of a service ☐

Submit result for a service ☐

» Hosts Actions Access

Enable/Disable Checks for a host ☐

Enable/Disable Notifications for a host ☐

Acknowledge a host ☒

Disacknowledge a host ☐

Schedule the check for a host ☒

Schedule the check for a host (Forced) ☒

Schedule downtime for a host ☒

Add/Delete a comment for a host ☒

Enable/Disable Event Handler for a host ☐

Enable/Disable Flap Detection for a host ☐

Enable/Disable Checks services of a host ☐

Enable/Disable Notifications services of a host ☐

Submit result for a host ☐

» Additional Information

Status ☒ Enabled ☐ Disabled

» List ☐ Form

Save | Reset

图 12-17 Centreon 的功能访问控制编辑页面

图 12-17 中，Global Functionalities Access（全局功能访问）选项组用以设定是否显示 Centreon 的 Web 页面顶端的计数器，如图 12-19 所示。Global Monitoring Engine Actions

(External Process Commands) (全局监控引擎选项(执行外部进程命令)) 选项设定是否允许用户向 Nagios 等调度引擎发出外部命令, 以执行重启、关闭、发送通知消息等功能。而 Services Actions Access (服务动作访问) 和 Hosts Actions Access (主机动作访问) 两个选项允许用户对主机和服务等监控资源执行告警确认、重新调度执行、添加备注等动作。

12.4.2 访问组

访问控制策略一旦被制订, 就会被归类到不同的访问组 (Access Group), 并且与用户组关联, 以将访问控制策略具体落实到不同的用户身上。为实现这一点, 需要选择菜单 Administration→ACL, 并单击左侧竖状菜单中的 Access Groups 子菜单, 进入访问组的列表与编辑界面, 如图 12-18 所示。

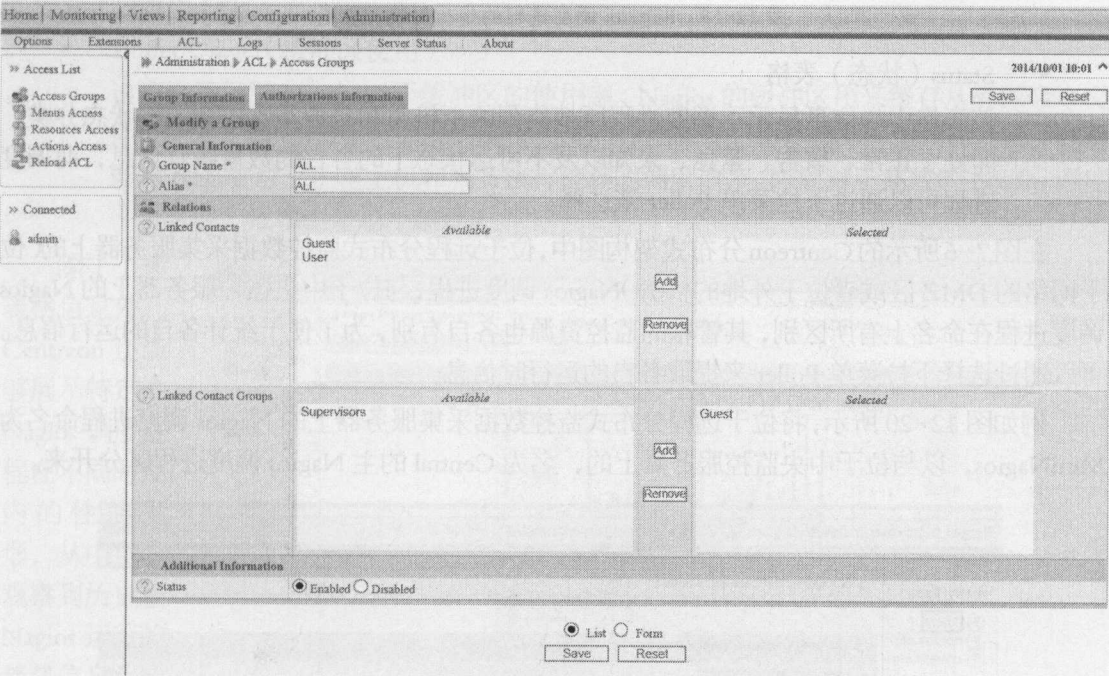


图 12-18 访问组的编辑界面

在图 12-18 中的 Group Information (组信息) 选项卡中, 可以指定该访问组的名称, 以及相关联的联系人或者联系人组, 当然也可以在联系人或者联系人组的编辑页面中指定这种关联关系, 如图 11-所示。接下来在 Authorizations Information (授权信息) 选项卡中, 可以依次选择资源访问控制项、菜单访问控制项、或者功能访问控制项, 单击 Save (保存) 链接, 即可与该访问组相关联。

12.5 调度进程的运行时统计信息

Centreon 会定期地统计调度进程, 即 Nagios 运行过程中的相关信息, 并且会显示在菜单 Home→Monitoring Engine Statistics中, 如图 12-19 所示。

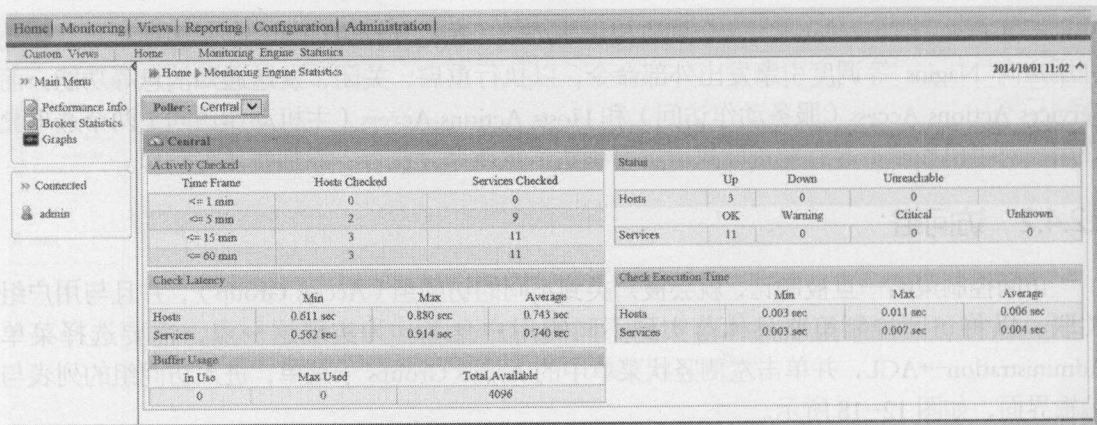


图 12-19 Nagios 调度进程的运行时统计信息

■ Status（状态）表格

该表格中的信息显示了调度进程所搜集到的主机类和服务类监控资源的状态分类，即处于正常、警告、紧急、未知以及不可达状态下的各自的数量统计信息，而调度进程可以通过下拉菜单 Poller 来选择。

在图 7-6 所示的 Centreon 分布式架构图中，位于远程分布式监控数据采集服务器上的（位于网络的 DMZ 区或者位于外地的机房）Nagios 调度进程，与位于中央监控服务器上的 Nagios 调度进程在命名上有所区别，其管辖的监控资源也各自有别，为了便于统计各自的运行信息。可以通过选择下拉菜单 Poller 来提取各自的运行时信息。

例如图 12-20 所示，将位于远程分布式监控数据采集服务器上的 Nagios 调度进程命名为 MiniNagios，以与位于中央监控服务器上的，名为 Central 的主 Nagios 调度进程区分开来。

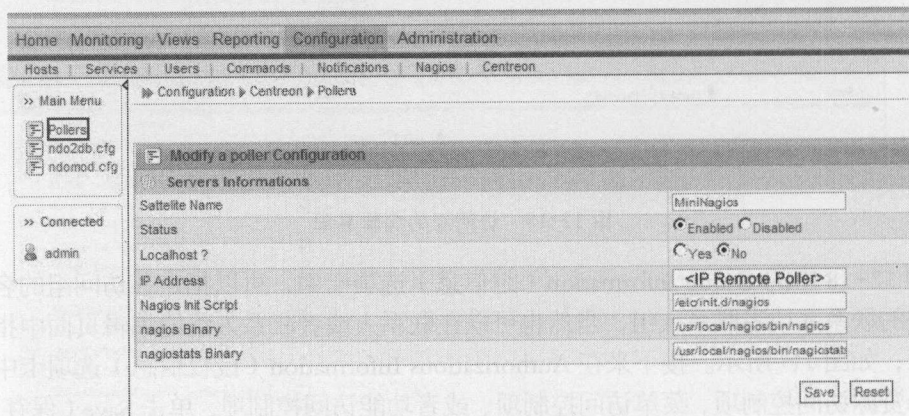


图 12-20 分布式 Nagios 调度进程的命名

■ Actively Checked（活动检查）表格

该表格显示了在不同的检查周期内，相应的 Nagios 调度进程所检测的主机以及服务的数量，这些指标有助于了解在单位时间内，Nagios 调度进程能够对多少数量的监控对象执行检测，或者有助于评估完成整个范围的对象检测需要花费多长时间。通过图 12-19 可以看出，Nagios 完成 3 台主机的检测所需时间要小于 15 分钟。

■ Check Execution Time (检测执行时间)

该列表提供了检测主机或者服务所需的最小检测时间、最大检测时间以及平均检测时间。在 Nagios 中，一般都会在全局配置文件中定义主机或者服务检测的最大允许时间，即超时时间，参考 9.2.3 小节。如果对于主机或者服务的检测时间超时，那么 Nagios 会将主机或者服务的状态标记为“未知”状态。

■ Check Latency (检测延迟)

该表格提供了主机和服务检测的相关延迟信息。所谓延迟，指的是主机和服务计划中的检测时间与实际执行检测时的时间之间的差值。对于性能突出的 Nagios 检测平台来说，这个延迟应该是无限趋近于 0。而如果观察到列表中的延迟信息较大，说明该 Nagios 调度进程对于主机和服务的检测效率很差劲，意味着后续需要进行相应的调优工作。

■ Buffer Usage (缓冲区使用)

该表格显示了 Nagios 对于缓冲区的使用率。Nagios 的缓冲区用来缓存从外部命令文件（参考 9.2.1 小节）中读取的高优先级的外部命令。如果缓冲区使用率过高，则会影响 Nagios 对于正常主机和服务执行检测的调度效率。因此对于高效的 Nagios 调度平台来说，该项值是越小越好。

除了上述表格信息外，Centreon 还能够展示特定的 Nagios 调度进程在不同时期内的性能图形，从中可以观察到历史上 Nagios 运行的趋势信息。该界面通过选择左侧竖状菜单中的 Graphs 项就可以进入，如图 12-21 所示。

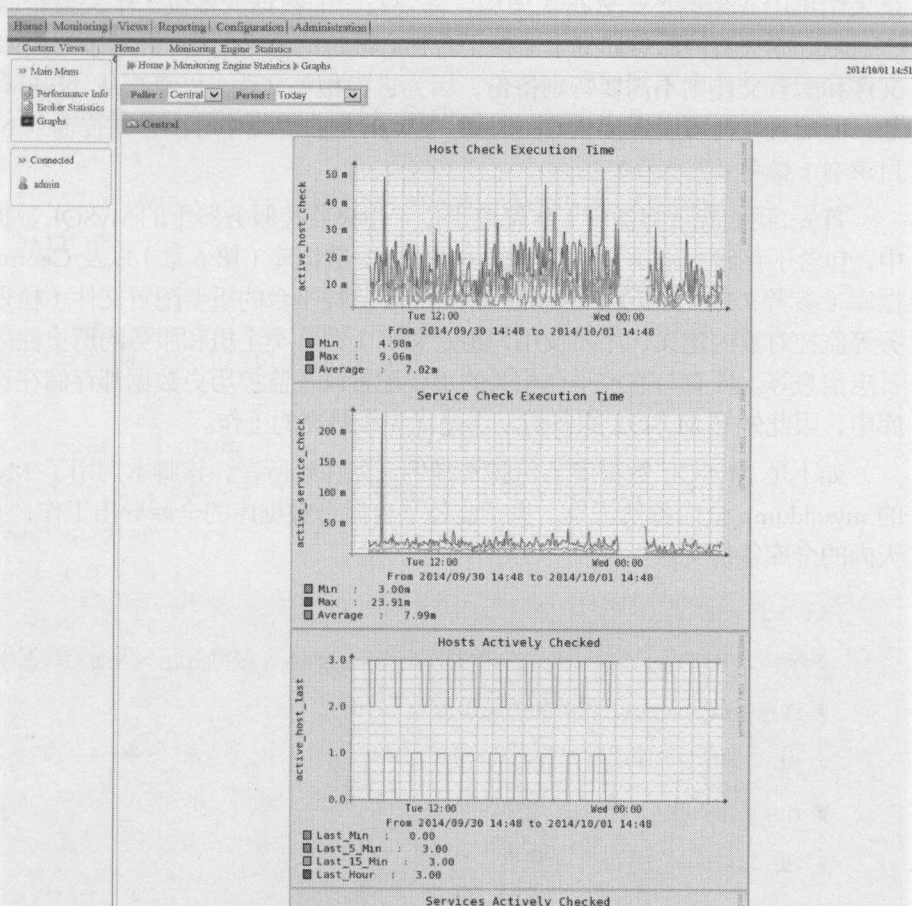


图 12-21 Nagios 的性能图形

12.6 Centreon 监控平台的备份与恢复

为保证对于企业级 IT 系统健康程度实施监控的效率和质量，并且保持高标准的报警效率，Centreon 监控平台自身也必须遵循企业级 IT 系统的一系列运维规范，包括 7*24 小时的不间断和高效率运行。为了达到这一目的，确保监控平台自身的系统备份以及应急情况下的系统恢复是非常有必要的。对于发生系统故障——无论是硬件故障还是软件故障的监控平台系统来说，尽快恢复其正常运行非常重要，因为监控平台的安全与其他系统的安全稳定息息相关，前者的故障可能导致后者的故障无法被及时观测到，从而引发更大范围的系统故障。从运行数据分析和预测的角度来说，及时恢复监控平台的系统故障并且保留历史监控数据，可以对系统累积的各类运维数据作容量分析，有助于后续的运行数据趋势分析和生成相关报表。当然，我们也要考虑到，在系统恢复的过程中，或许不可避免地会产生一些监控历史数据的丢失，从而影响监控数据的连贯性，因此，我们更应该设计出良好的备份与恢复策略，以及各类高可用性的方案，尽可能避免系统的失效和数据的丢失。

12.6.1 系统备份

提到备份，首要考虑的就是需要备份的对象及内容。对于 Centreon 监控平台来说，无论是采用集中式架构还是分布式架构，需要备份的数据大多位于中央监控服务器上，这也是 Centreon 中央监控服务器处于监控平台核心地位的意义所在。而其余的例如 Nagios 相关进程文件和配置文件则不需要特别备份，因为进程相关文件可以直接从 Nagios 的官方网站上下载，而配置文件可以直接从 Centreon 的 Web 用户界面中导出，并传输到 Nagios 的配置文件目录中（参考 8.7 小节）。

首先，我们应该将备份工作聚焦于位于中央监控服务器上的 MySQL 数据库上。在 MySQL 中，包含了存储 Nagios 相关数据的 NDOUtils 数据库（第 6 章）以及 Centreon 相关数据的数据库（参考 7.5.2 小节），包含了 Centreon 监控平台的重要配置文件（检查命令、主机和服务等监控对象的定义、用户及用户组定义等），各类主机和服务的历史性能数据，以及相关日志信息等。所有与监控平台相关的重要配置以及监控历史数据都存储在这个庞大的数据仓库中，因此做好 MySQL 的备份工作是优先级最高的工作。

如下是 MySQL 数据库备份脚本样例，如注释所言，该脚本调用了 MySQL 数据库提供的 mysqldump 全库备份工具，执行监控平台后台数据库的全库导出工作，并能够保留最近 5 天内的全库备份文件。

```
#!/bin/sh

#mysql_backup.sh: MySQL 数据库全库备份脚本，保留最近 5 天内的数据库全库备份文件

# 注意参数为 MySQL 相关的登录信息

# db_user : mysql 用户名

# db_passwd : mysql 密码

# db_host : mysql 主机名

# -----
```



```

db_user="root"
db_passwd="root"
db_host="localhost"
# the directory for store your backup file.
backup_dir="/mysqlbackup"
# date format for backup file (dd-mm-yyyy)
time="$(date +%d-%m-%Y)"
# mysql, mysqldump and some other bin's path
MYSQL="/usr/bin/mysql"
MYSQLDUMP="/usr/bin/mysqldump"
MKDIR="/bin/mkdir"
RM="/bin/rm"
MV="/bin/mv"
GZIP="/bin/gzip"
# check the directory for store backup is writeable
test ! -w $backup_dir && echo "Error: $backup_dir is un-writeable." && exit
0
# the directory for store the newest backup
test ! -d "$backup_dir/backup.0/" && $MKDIR "$backup_dir/backup.0/"
# get all databases
all_db="$($MYSQL -u $db_user -h $db_host -p$db_passwd -Bse 'showdatabases') "
for db in $all_db
do
$MYSQLDUMP -u $db_user -h $db_host -p$db_passwd $db | $GZIP -9 >
"$backup_dir/backup.0/$time.$db.gz"
done
# delete the oldest backup
test -d "$backup_dir/backup.5/" && $RM -rf "$backup_dir/backup.5"
# rotate backup directory
for int in 4 3 2 1 0
do
if(test -d "$backup_dir/backup.$int")
then
next_int=`expr $int + 1`
$MV "$backup_dir/backup.$int" "$backup_dir/backup.$next_int"
fi
done
exit 0;

```


接下来，我们应注意到检测命令的备份工作。不仅仅是因为检测命令的丰富多样（往往因监控平台所运行操作系统及运行环境版本的不同而经过重新配置或编写，甚至每个监控平台的检测命令都有版本差异），更重要的是需要备份那些我们自定义的，非系统默认提供的检测命令，这样在系统恢复后才不至于重新配置或编写这些命令。另外一点需要注意的是，这些检测命令所运行的环境同样需要备份，例如 Perl 运行环境、GCC 运行环境、Python 运行环境等。

最后，我们需要考虑的是备份文件系统，包括 Nagios 的运行目录、Centreon 的运行目录、NagVis 的运行目录、MySQL 的运行目录、Apache 的运行目录等重要组件的文件系统。这些文件系统中包含有重要的配置文件，且体积庞大，因此需要进行压缩备份。

以下是监控平台重要文件系统以及相关检测命令的备份和压缩脚本样例：

```
#!/bin/bash
#该脚本用以备份 Nagios、NagVis 相关运行目录,监控探针以及 Centreon 相关运行目录
#定义监控服务器 IP
MONITOR="xxx.xxx.xxx.xxx"#监控平台虚拟 IP

db_user=root
db_pass=xxxxxx
db_name=centreon
nagios_config=/usr/local/nagios/etc
nagios_libexec=/usr/local/nagios/libexec
nagvis_dir=/usr/local/nagios/nagvis
centreon_dir=/usr/local/centreon
nagios_bak_dir=/nagios_bak/backup_os_config
nagios_back_dir=/nagios_bak
log_file=/nagios_bak/nagios_backup_cron.log
cur_time=$( date +%Y%m%d )

#####
# 判断运行环境，保证是 nagios 服务器 #
#####
ip addr |grep "$MONITOR" >/dev/null
Real=$?
if [[ "$Real" != "0" ]];then
    echo "${cur_time} - This machine is not real nagios server." >> $log_file
    exit
fi
```

```

#首先进入用以存放备份文件的目录
cd ${nagios_bak_dir}

#备份 nagios 配置文件，并存放备份目录下，其余备份均采用同样操作
tar zcvf nagios_etc_${cur_time}.tar.gz ${nagios_config}
if [[ $? -eq 0 ]];then
    echo "${cur_time} - Backup nagios etc OK" >>${log_file}
else
    echo "${cur_time} - Backup nagios etc Err" >>${log_file}
fi

#备份 nagios 检测命令文件
tar zcvf nagios_libexec_${cur_time}.tar.gz ${nagios_libexec}
if [[ $? -eq 0 ]];then
    echo "${cur_time} - Backup nagios libexec OK" >>${log_file}
else
    echo "${cur_time} - Backup nagios libexec Err" >>${log_file}
fi

#备份 nagvis 运行目录
tar zcvf nagvis_${cur_time}.tar.gz ${nagvis_dir}
if [[ $? -eq 0 ]];then
    echo "${cur_time} - Backup nagvis OK" >>${log_file}
else
    echo "${cur_time} - Backup nagvis Err" >>${log_file}
fi

#备份 centreon 运行目录
tar zcvf centreon_${cur_time}.tar.gz ${centreon_dir}
if [[ $? -eq 0 ]];then
    echo "${cur_time} - Backup centreon OK" >>${log_file}
else
    echo "${cur_time} - Backup centreon Err" >>${log_file}
fi

```

上述备份工作完成后，还需要将备份文件拷贝传输到专用的备份服务器上，避免单台服务器产生故障影响备份文件的有效性。

以下脚本样例用于将位于本地备份目录下的备份文件传送至远程服务器相关目录:

```
#!/bin/bash
MONITOR="xxx.xxx.xxx.xxx"
MONITOR_BAK="xxx.xxx.xxx.xxx"

db_user=root
db_pass=***
db_name=centreon
nagios_config=/usr/local/nagios/etc
nagios_libexec=/usr/local/nagios/libexec
nagvis_dir=/usr/local/nagios/nagvis
centreon_dir=/usr/local/centreon
nagios_bak_dir=/nagios_bak/backup_os_config
nagios_back_dir=/nagios_bak
log_file=/nagios_bak/nagios_backup_cron.log
cur_time=$( date +%Y%m%d )

#####
# 判断运行环境, 保证是 nagios 服务器 #
#####
ip addr |grep "$MONITOR" >/dev/null
Real=$?
if [[ "$Real" != "0" ]];then
    echo "${cur_time} - This machine is not real nagios server." >> $log_file
    exit
fi

#使用 scp 命令, 将备份文件由主服务器传送至远程服务器上的备份目录
scp ${nagios_bak_dir}/nagios_etc_${cur_time}.tar.gz
    ${MONITOR_BAK}:${nagios_back_dir}
scp1=$?

scp ${nagios_bak_dir}/nagios_libexec_${cur_time}.tar.gz
    ${MONITOR_BAK}:${nagios_back_dir}
scp2=$?

scp ${nagios_bak_dir}/nagvis_${cur_time}.tar.gz
    ${MONITOR_BAK}:${nagios_back_dir}
scp3=$?

scp ${nagios_bak_dir}/centreon_${cur_time}.tar.gz
    ${MONITOR_BAK}:${nagios_back_dir}
```



```

scp4=$?
scp /mysqlbackup/backup.1/*.gz ${MONITOR_BAK}:${nagios_back_dir}
scp5=$?
if [ $scp1 -eq 0 ] && [ $scp2 -eq 0 ] && [ $scp3 -eq 0 ] && [ $scp4 -eq 0 ]
  && [ $scp5 -eq 0 ];then
    echo "${cur_time} - SCP BAK to MONITOR_BAK OK" >>${log_file}
else
    echo "${cur_time} - SCP BAK to MONITOR_BAK Err" >>${log_file}
fi
#删除本地 3 天前的备份文件
find ${nagios_bak_dir} -mtime +3 -name "nagios_etc*" -exec rm -f {} \;
find ${nagios_bak_dir} -mtime +3 -name "nagios_libexec*" -exec rm -f {} \;
find ${nagios_bak_dir} -mtime +3 -name "centreon*" -exec rm -f {} \;

```

在实际运行中,我们往往在操作系统添加定时任务,调用脚本,在夜间完成数据库备份、文件系统备份,以及传输到另外一台服务器的所有工作,步骤如下:

- (1) 将脚本存放至监控服务器的本地目录,例如/usr/local/bin下;
- (2) 使用 root 用户,执行 vi /etc/crontab 命令,定义定时任务;
- (3) 添加如下定时任务:

```
00 03 1 * * root /usr/local/bin/nagios_bakcup.sh >/dev/null 2>&1
```

上述定时任务表明在每月1号的凌晨3点执行 Nagios 备份作业,而接下来的其他定时任务可根据策略自由定义。请注意上述脚本仅为样例,在本书中作为示例所用。如需用于生产系统,则需要经过一定量的定制开发,经详细测试后才能部署运行。

12.6.2 系统恢复

监控系统恢复的步骤,大体上是备份的步骤反过来的样子。一般先恢复 MySQL 数据库,除了要安装合适的 MySQL 数据库版本外(注意操作系统以及各类软件的兼容程度),还需要用备份出来的 MySQL 数据库备份文件恢复出 NDOUtils 数据库以及 Centreon 相关数据库,并创建用户和赋予相应权限。之后就是各类组件运行目录的恢复过程,恢复完毕后,要注意检查各自文件系统的权限和属主属组等属性。完成上述工作后,还需要检查各类插件运行环境,例如 Perl 运行库、Python 运行库以及特殊插件需要用到的运行库等是否齐备,使插件能够正常运行。

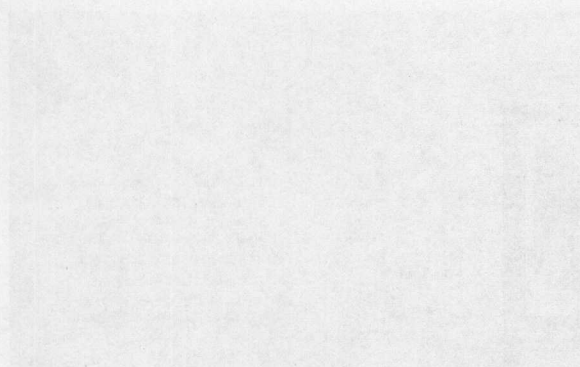
完成上述工作后,在检查 IP 地址和网络配置正确的情况下,可以启动监控平台的各类组件,尝试打开浏览器,在 Centreon 监控平台的 Web 用户界面中查看各类监控项、用户、权限、检查命令等配置是否已经正常恢复,各类监控对象的状态是否实时,是否不再全部处于 Critical 状态或者 Unknown 状态。并且可以尝试更改一些配置,观察是否生效,以此判断整个 Centreon 监控平台的功能是否已经完全在备机上运作正常。



第 13 章

NagVis 的安装与配置

在 1.5 小节中已经介绍到，NagVis 是 Nagios 的著名图形插件，用以在一张背景地图上显示被监控对象的状态信息。无论是主机组、服务组、主机或是服务，都能够以图标的形式被摆放于地图上。不同的图标对应着监控对象的不同状态，而地图则取决于系统管理员的偏好，可以是一张实际的机房地图，或者系统的逻辑架构图，甚至是一张机柜图片。



13.1 关于 NagVis

用以代表监控对象的图标可以自由摆放在地图的不同位置，以代表不同的物理监控对象，例如服务器、磁盘，电源等，还可以代表逻辑监控对象，例如网络端口、交易数量，以及 CPU 利用率等。在显示监控对象状态的时候，NagVis 依然遵循着状态优先级的原则，优先显示处于 Critical（紧急）状态下的对象，接着是 Warning（警告）状态、Unknown（未知）状态，最后，OK（正常）状态的优先级为最低。

例如，某个监控图标代表某项主机组，如果该组中某台主机处于 Warning（警告）状态而其他主机处于 OK（正常）状态，那么该监控图标即处于 Warning（警告）状态；如果该主机组中的其他任何一台主机转换为 Critical（紧急）状态，那么该图标将立即转换为 Critical（紧急）状态，因为根据状态优先级的原则，Critical（紧急）状态为最高优先级的状态，应该优先显示，这就意味着处于 Critical（紧急）状态下的监控项应该优先被关注。

13.1.1 地图关系设定

除了状态优先级之外，NagVis 的另外一个特色是支持地图之间的父子关系设定，一张设计好的地图可作为图标的形式，放置在另外一张地图中，使得在一块大屏幕上展示大范围的网路拓扑图成为可能。例如，用户可以设计一张数据中心建筑图作为背景，其上摆放各个机房视图图标，单击各个机房图标，用户的浏览器可以跳转到对应的机房背景图中，从而定位到故障服务器。

如图 13-1 所示，指定“2 层机房 1 号机柜（其同义词是‘2 层机房’）”地图的父地图为“serverRoom”。

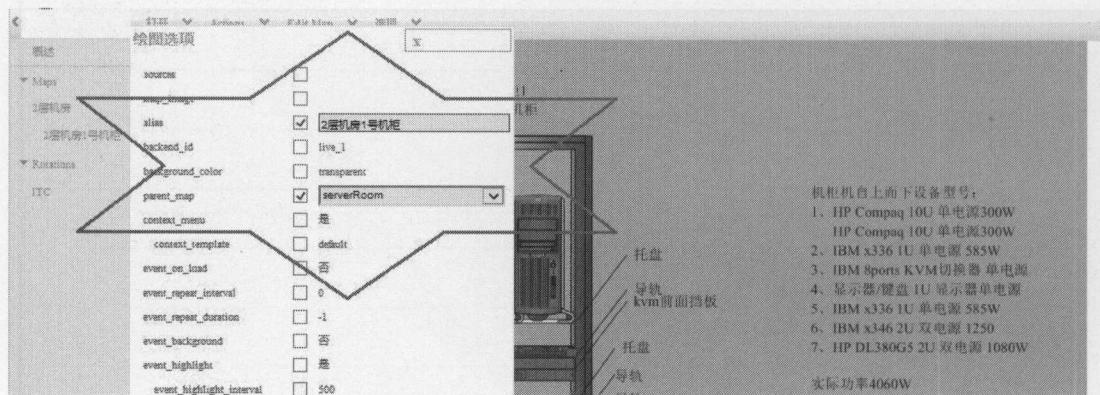


图 13-1 NagVis 中“父-子”地图关系的设定

在地图列表上可以看到，地图“2 层机房（为 serverRoom 地图的中文同义词）”与“2 层机房 1 号机柜”为层级父子关系，如图 13-2 所示。

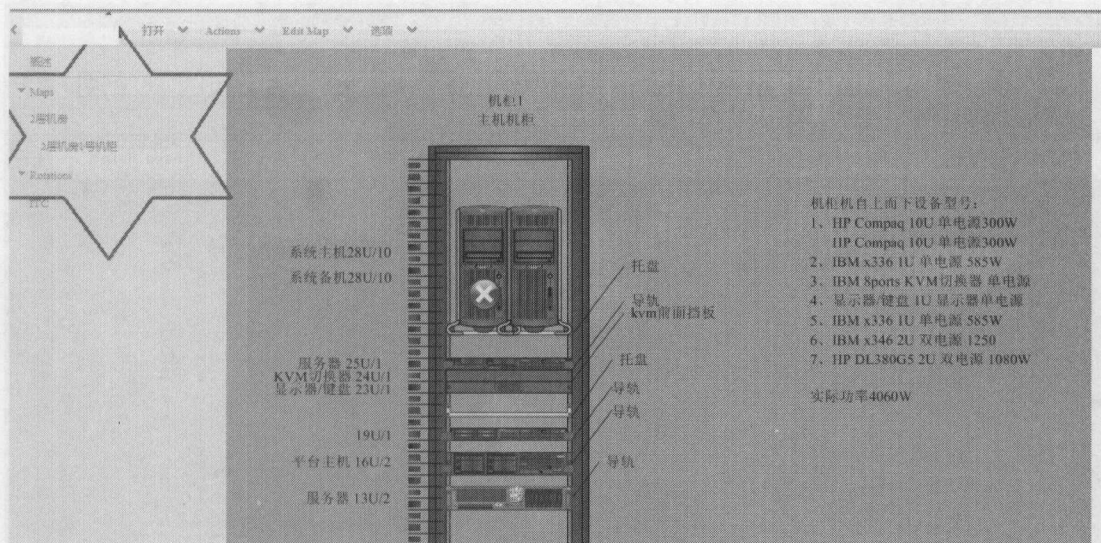


图 13-2 NagVis 中“父-子”地图关系的显示

而当子地图“2层机房 1 号机柜”的状态为 Critical（紧急）状态时，父地图“2 层机房”的状态同样显示为 Critical（紧急）状态，如图 13-3 所示。

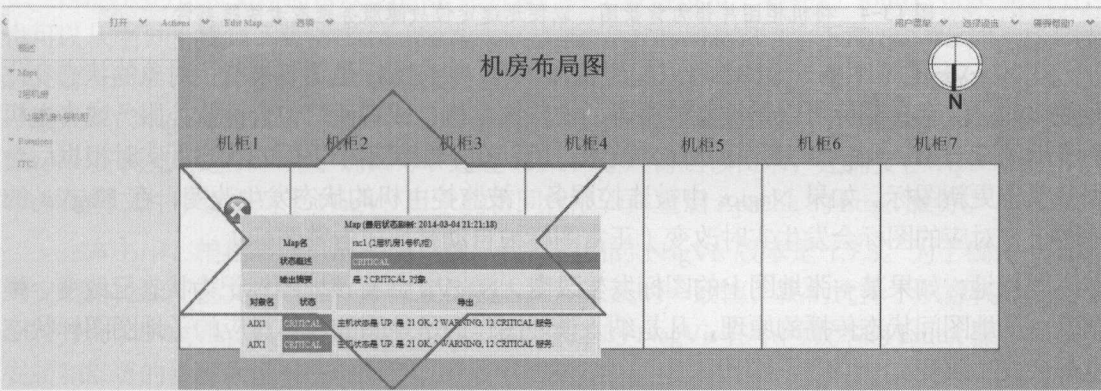


图 13-3 NagVis 中“父”地图显示“子”地图状态

13.1.2 NagVis 的地图

具体说来，NagVis 的地图通常是由一张背景图片以及一系列代表被监控主机或者被监控服务的图标组成。与生活中常见的实际地图类似，NagVis 既可以用一张实际的网络部署图片或者位于不同地理位置的实际机房图片作为地图，也可引入网络的逻辑架构图作为背景地图。在 NagVis 的概念中，只要是能够有助于监控人员理解并定位故障主机或者故障服务位置的图片，都可以作为背景地图出现。如图 13-4 所示，显示了一张以实际机柜图片作为监控地图的样例。

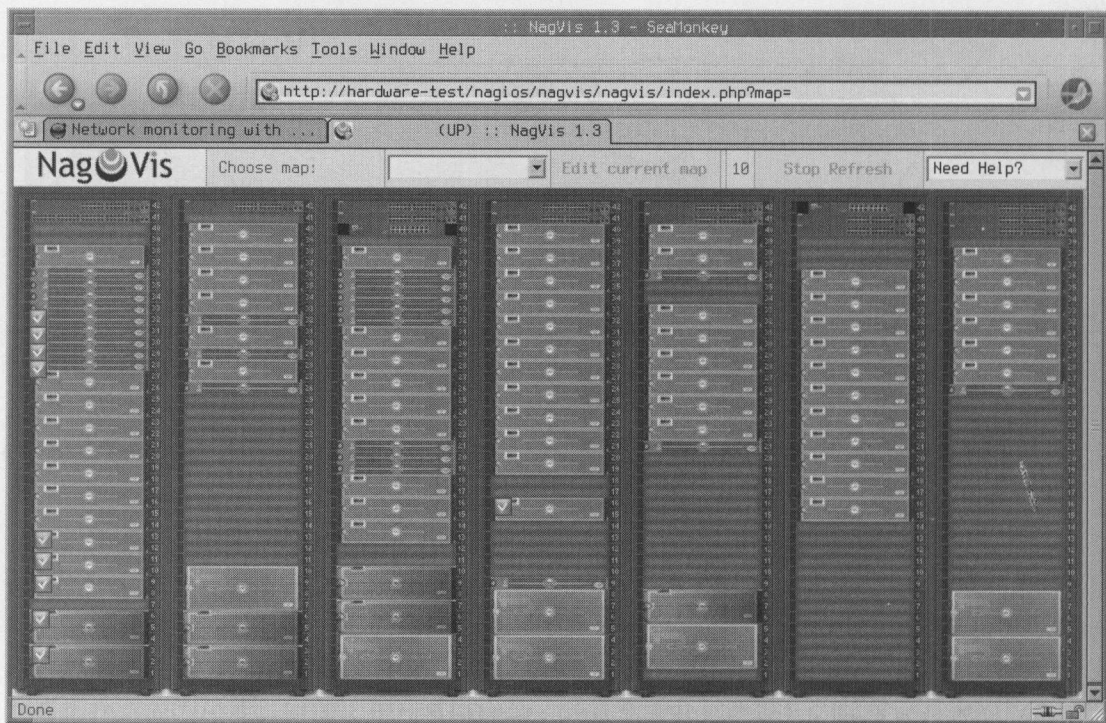


图 13-4 将机柜图片作为背景图，以便迅速定位故障服务器或者故障部件

在 NagVis 的地图中，背景图片只起到装饰作用，唯有图标是随着被监控对象的状态变化而变化的。注意，图标可以是被监控服务、被监控主机、或者任何包含被监控服务或者主机的子地图等对象。NagVis 采用了 AJAX 技术，可以在不刷新整个页面的状态下实时根据后台服务状态更新图标。如果 Nagios 中被监控服务、被监控主机的状态发生改变，在 NagVis 的地图上，对应的图标会发生实时改变（正常图标被自动替换成告警图标）。

同样地，如果某一张地图上的图标发生改变，NagVis 会将该地图标记为状态已改变，根据父-子地图间状态传播的原理，凡是纳入该地图作为子地图的，其对应的子地图图标状态也都会发生改变。

13.2 NagVis 的运作机制

NagVis 是一个基于 PHP 和 AJAX 技术构建的 Web 应用，是 Nagios 的插件，这就意味着必须依赖于 Nagios 才能展示数据。只有当系统安装并配置了 Nagios、MySQL 数据库以及 Nagios 的检测数据输出工具 NDOUtils（NDOUtils 是 Nagios 的官方数据库接口组件，由 Nagios 的开发者 Ethan Galstad 编写，可以将一个或多个 Nagios 实例采集到的实时及历史监控数据导出到 MySQL 数据库中，请参考第 6 章的内容），NagVis 才能访问并展示这些监控信息。换句话说，NagVis 必须依靠 NDOUtils 作为自身的数据源。

NagVis 是一个基于 Web 的应用，因此需要 Apache HTTP Server 和至少 PHP 4.2 以上版本的运行环境。另外，NagVis 的宿主 Linux 操作系统上的一些运行时系统软件包也需要安装，

这些包在安装操作系统时并非默认安装,但往往包含有 NagVis 运行时所需要的系统文件。这就意味着,如果安装或运行 NagVis 的过程中,可能会出现某些库文件的提示,一旦遇到此类提示,就需要寻找并安装相应版本的依赖文件,才能继续后续的工作。

尽管 NagVis 和 Nagios 是共生关系,但并不意味着 NagVis 必须和 Nagios 运行在同一台服务器上,相反地,它们可以分开部署。例如,可以采取将 NagVis 服务器部署在企业内网的 DMZ 区域,将 Nagios 部署在企业内网的网管网区域的方式。顺便提一下,DMZ 是英文 demilitarized zone 的缩写,中文名称为“隔离区”,也称“非军事化区”。DMZ 区是为了解决安装防火墙后外部网络不能访问内部网络服务器的问题,而设立的一个非安全系统与安全系统之间的缓冲区,这个缓冲区位于企业内部网络和外部网络之间的小网络区域内,在这个小网络区域内可以放置一些必须公开的服务器设施,如企业 Web 服务器、FTP 服务器和论坛等。另一方面,通过这样一个 DMZ 区域,更加有效地保护了内部网络,因为这种网络部署,比起一般的防火墙方案,对攻击者来说又多了一道关卡。

如此一来,内网的服务器信息经网管网中的 Nagios 服务器采集后,再送达可供提供外部用户访问服务的 NagVis 服务器上展示,比起这两者部署在同一台服务器的架构来说,更有效地提升了系统的安全性和健壮性。

13.3 NagVis 的安装

NagVis 的安装文件通常以 gzip 格式封装,其下载链接位于著名的 SourceForge 网站上,也可以从主页上找到下载链接(<http://www.nagvis.org/downloads>)。下载 NagVis 的 gzip 安装包并解压缩后,可以看到 NagVis 的文件目录以 nagvis-版本号的形式命名,例如 nagvis-1.7.3。一般来说,NagVis 的默认安装路径是安装到/usr/local 目录下,例如,至/usr/local/nagvis 目录。NagVis 安装完毕之后,为了确保可以通过 HTTP 方式访问该目录,还需要在 Apache 的相关 Web 配置文件中设置对于 NagVis 目录的访问权限,并重启 Apache 的 httpd 服务。

在本书中,根据表 5-1 的安装规划,我们选用的 NagVis 版本是 1.7.3。为了确保 NagVis 的安装路径和配置文件路径在不同发行版本操作系统的一致性,我们选择了从源代码开始中配置并安装 NagVis。只要遵从以下步骤,不同版本的 NagVis 在不同版本的 Linux 操作系统上安装和部署的步骤就没有任何差别。

接下来的步骤中,我们将下载到的 nagvis-1.7.3.tar.gz 文件放到/tmp 目录下,执行解压缩的步骤。

```
[root@monitor tmp]# tar xvzf nagvis-1.7.3.tar.gz
```

解压后的的 NagVis 以目录 nagvis-1.7.3 的形式存放在/tmp 文件系统中,接下来我们使用 ls 命令查看该目录的结构:

```
[root@monitor local]# ls -F /tmp/nagvis-1.7.3
ChangeLog  INSTALL      LICENCE      pack*  TODO
docs/      install_lib  nagvis-make-admin*  README  uifx/
etc/       install.sh*  omd_install.sh*  share/
```

通过 ls 命令可以看到 NagVis 的目录下存在 install.sh 安装文件，接下来需要将该安装文件的权限设置为可执行：

```
[root@monitor nagvis-1.7.3]# chmod +x install.sh
```

接下来运行 install.sh 文件，执行 NagVis 的安装步骤：

```
[root@monitor nagvis-1.7.3]# ./install.sh
```

执行 install.sh 安装文件后，首先展现 NagVis 的欢迎界面，在 Do you want to proceed?（是否希望安装步骤继续）中填写“y”，以执行下一阶段安装步骤。系统默认会识别出 Nagios 的安装路径为/usr/local/nagios，并且将 NagVis 安装在路径/usr/local/下，与 Nagios 所在目录一致。

```
+-----+
| Welcome to NagVis Installer 1.7.3                                     |
+-----+
| This script is built to facilitate the NagVis installation and update   |
| procedure for you. The installer has been tested on the following systems:|
| - Debian, since Etch (4.0)                                             |
| - Ubuntu, since Hardy (8.04)                                          |
| - SuSE Linux Enterprise Server 10 and 11                               |
|                                                                         |
| Similar distributions to the ones mentioned above should work as well. |
| That (hopefully) includes RedHat, Fedora, CentOS, OpenSuSE           |
|                                                                         |
| If you experience any problems using these or other distributions, please |
| report that to the NagVis team.                                       |
+-----+
| Do you want to proceed? [y]: y                                         |
+-----+
| Starting installation of NagVis 1.7.3                                   |
+-----+
| OS : Red Hat Enterprise Linux Server release 6.5 (Santiago)           |
```



```

|
|
+--- Checking for tools -----+
| Using packet manager /bin/rpm                                found |
|
|
+--- Checking paths -----+
| Please enter the path to the nagios base directory [/usr/local/nagios]:
|   nagios path /usr/local/nagios                                found |
| Please enter the path to NagVis base [/usr/local/nagvis]:
|
|

```

接下来,安装脚本会检测操作系统是否存在所依赖的软件包。为使 NagVis 软件能够正常运转,脚本所列举的所有软件包都应该是 found (存在) 状态,如果检测过程中出现缺失的软件包,就需要检查并安装缺失组件,并再次运行 `install.sh` 脚本,以再次检测组件并安装。

在 NagVis 的安装过程中,基本上不需要用户填写任何路径信息和配置信息,在遇到系统安装参数提示信息时,按照系统提供的默认安装参数,直接按回车键确认即可。

```

+--- Checking prerequisites -----+
| PHP 5.3                                                        found |
|   PHP Module: gd php                                          found |
|   PHP Module: mbstring php                                    found |
|   PHP Module: gettext compiled_in                             found |
|   PHP Module: session compiled_in                             found |
|   PHP Module: xml php                                         found |
|   PHP Module: pdo php                                         found |
|   Apache mod_php                                              found |
| Do you want to update the backend configuration? [n]:
|   Graphviz 2.26                                               found |
|   Graphviz Module dot 2.26.0                                  found |
|   Graphviz Module neato 2.26.0                                found |
|   Graphviz Module twopi 2.26.0                                found |
|   Graphviz Module circo 2.26.0                                found |

```



```

| Graphviz Module fdp 2.26.0                                found |
| SQLite 3.6                                                found |
|                                                           |
+--- Trying to detect Apache settings -----+
| Please enter the web path to NagVis [/nagvis]:
| Please enter the name of the web-server user [apache]:
| Please enter the name of the web-server group [apache]:
| create Apache config file [y]:
|                                                           |
+--- Checking for existing NagVis -----+
| NagVis 1.7.3                                              found |
| Do you want the installer to update your config files when possible? [y]:
| Remove backup directory after successful installation? [n]:
|                                                           |
+-----+
| Summary                                                    |
+-----+
| NagVis home will be:          /usr/local/nagvis          |
| Owner of NagVis files will be: apache                    |
| Group of NagVis files will be: apache                    |
| Path to Apache config dir is: /etc/httpd/conf.d          |
| Apache config will be created: yes                        |
|                                                           |
| Installation mode:          update                        |
| Old version:                1.7.3                        |
| New version:                1.7.3                        |
| Backup directory:           /usr/local/nagvis.old-2014-10-20_11:08:30 |
|                                                           |
|Note: The current NagVis directory will be moved to the backup directory. |

```

```

|The backup directory will be NOT removed after successful installation |
|
|Your configuration files will be copied. |
|
|The configuration files will be updated if possible. |
|
|
| Do you really want to continue? [y]:
+-----+
| Starting installation |
+-----+
| Moving old NagVis to /usr/local/nagvis.old-2014-10-20_11:08:30.. done |
| Creating directory /usr/local/nagvis... done |
| Creating directory /usr/local/nagvis/var... done |
| Creating directory /usr/local/nagvis/var/tmpl/cache... done |
| Creating directory /usr/local/nagvis/var/tmpl/compile... done |
| Creating directory /usr/local/nagvis/share/var... done |
| Copying files to /usr/local/nagvis... done |
| Creating directory /usr/local/nagvis/etc/profiles... done |
| Creating main configuration file... done |
| Adding webserver group to file_group... done |
| Creating web configuration file... done |
| Setting permissions for web configuration file... done |
|
|
| Restoring custom map configuration files... done |
| Restoring custom geomap source files... done |
| Restoring conf.d/ configuration files... done |
| Restoring custom map images... done |
| Restoring custom gadget images... done |
| Restoring custom iconsets... done |
| Restoring custom shapes... done |
| Restoring custom templates... done |

```



```

| Restoring custom template images... done |
| Restoring custom gadgets... done |
| Restoring custom scripts... done |
| Restoring custom stylesheets... done |
| |
+-----+
| Handling changed/removed options |
+-----+
| Removing allowedforconfig option from main config... done |
| Removing autoupdatefreq option from main config... done |
| Removing htmlwuijs option from main config... done |
| Removing wuijs option from main config... done |
| Removing showautomaps option from main config... done |
| Removing usegdlbjs option from main config... done |
| Removing displayheader option from main config... done |
| Removing hovertimeout option from main config... done |
| Removing allowed_for_config option from map configs... done |
| Removing allowed_user from map configs... done |
| Removing hover_timeout from map configs... done |
| Removing usegdlbjs from map configs... done |
+-----+
| HINT: Please check the changelog or the documentation for changes which |
|       affect your configuration files |
| |
+--- Setting permissions... -----+
| /usr/local/nagvis/etc/nagvis.ini.php-sample done |
| /usr/local/nagvis/etc done |
| /usr/local/nagvis/etc/maps done |
| /usr/local/nagvis/etc/maps/* done |

```



```

| /usr/local/nagvis/etc/geomap                                done |
| /usr/local/nagvis/etc/geomap/*                              done |
| /usr/local/nagvis/etc/profiles                              done |
| /usr/local/nagvis/share/userfiles/images/maps              done |
| /usr/local/nagvis/share/userfiles/images/maps/*            done |
| /usr/local/nagvis/share/userfiles/images/shapes             done |
| /usr/local/nagvis/share/userfiles/images/shapes/*          done |
| /usr/local/nagvis/var                                       done |
| /usr/local/nagvis/var/*                                     done |
| /usr/local/nagvis/var/tmpl                                  done |
| /usr/local/nagvis/var/tmpl/cache                            done |
| /usr/local/nagvis/var/tmpl/compile                          done |
| /usr/local/nagvis/share/var                                  done |
|                                                              |
+-----+
| Installation complete                                         |
|                                                              |
| You can safely remove this source directory.                 |
|                                                              |
| For later update/upgrade you may use this command to have a faster update: |
| ./install.sh -n /usr/local/nagios -p /usr/local/nagvis -u apache -g apache -w |
/etc/httpd/conf.d -a y
|                                                              |
| What to do next?                                             |
| - Read the documentation                                     |
| - Maybe you want to edit the main configuration file?       |
|   Its location is: /usr/local/nagvis/etc/nagvis.ini.php     |
| - Configure NagVis via browser                               |
| <http://localhost/nagvis/config.php> |

```

```
| - Initial admin credentials: |
|
|   Username: admin           |
|
|   Password: admin          |
|
+-----+

```

安装脚本在执行过程中，除了会安装 NagVis 的运行文件和配置文件外，还能够自动产生 Apache Web 服务器的模块配置文件 `nagvis.conf`，存放在路径 `/etc/httpd/conf.d` 下。此举是为了保持不同模块的配置文件，例如 `Nagios.conf`（参看 5.5 下节）等位于 `/etc/httpd/conf.d` 目录下的后缀为 `.conf` 的配置文件，与位于 `/etc/httpd/conf/` 目录下的 Apache 原始配置文件 `httpd.conf` 之间的独立。

安装完毕后，使用 `service httpd restart` 命令重启 Apache 服务，使其能够识别新增的 NagVis 模块。打开浏览器，输入 `http://your-monitor-ip/nagvis`，即可访问 NagVis，登录用户名和密码分别是 `admin/admin`，如图 13-5 所示。

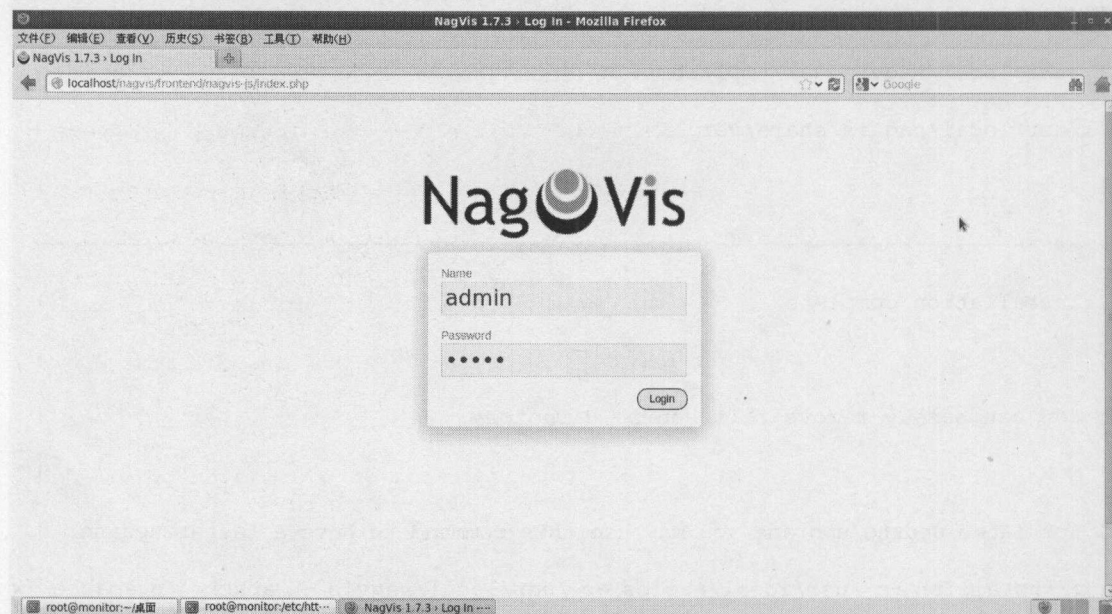


图 13-5 访问 NagVis

13.4 NagVis 的配置

继续 NagVis 的探索之旅，接下来让我们查看一下 NagVis 极其重要的配置文件目录——`etc` 目录。当首次安装完毕之后，在 NagVis 的 `etc` 目录（路径通常为 `/usr/local/nagvis/etc`）中，可以找到名为 `nagvis.ini.php-sample` 的样例配置文件，注意该文件只是 NagVis 的配置文件模板，并非真正有效的配置文件。下一步我们需要做的就是将 `nagvis.ini.php-sample` 文件重命名为 `nagvis.ini.php` 文件，使其成为真正的配置文件，之后在该文件基础上做必要的修改（别忘了修改前做好该文件的备份），如图 13-6 所示。


```

[root@monitorbak etc]# ls -ltr
总用量 112
drwxr-xr-x 2 apache apache 4096 2月 27 2013 profiles
drwxr-xr-x 2 apache apache 4096 2月 27 2013 geomap
drwxr-xr-x 2 apache apache 4096 2月 27 2013 conf.d
-rwxr-xr-x 1 apache apache 18102 2月 27 2013 nagvis.ini.php-sample
-rwxr-xr-x 1 apache apache 18447 2月 27 2013 nagvis.ini.php.bak
-rwxr-xr-x 1 apache apache 30 2月 27 2013 .htaccess
-rwxr-xr-x 1 apache apache 2338 2月 27 2013 apache2-nagvis.conf-sample
drwxr-xr-x 5 apache apache 4096 3月 2 2013 .
drwxr-xr-x 2 apache apache 4096 3月 4 21:49 maps
-rwxr-xr-x 1 apache apache 19456 3月 4 21:49 auth.db
-rw-rw---- 1 apache apache 18522 3月 5 03:42 nagvis.ini.php
drwxr-xr-x 6 apache apache 4096 3月 5 03:42 .
[root@monitorbak etc]# pwd
/ust/local/nagios/nagvis/etc

```

图 13-6 位于 nagvis/etc 目录下的 nagvis.ini.php 文件

我们可以用操作系统自带的文本编辑器打开并编辑 `nagvis.ini.php` 文件，该文件看起来与普通的 Windows 操作系统上的 `ini` 配置文件类似——配置文件中的不同模块以方括号显示，行首的分号代表该文本行为注释信息，每个配置项都以“属性=值”的方式表示。例如，默认的全局配置模块如下所示：

```

[global]

; Enable/Disable logging of security related user actions in Nagvis. For
; example user logins and logouts are logged in var/nagvis-audit.log
;audit_log="1"

;

; Defines the authentication module to use. By default NagVis uses the built-in
; SQLite authentication module. On delivery there is no other authentication
; module available. It is possible to add own authentication modules for
; supporting other authorisation mechanisms. For details take a look at the
; documentation.
;authmodule="CoreAuthModSQLite"

;

; Defines the authorisation module to use. By default NagVis uses the built-in
; SQLite authorisation module. On delivery there is no other authorisation
; module available. It is possible to add own authorisation modules for
; supporting other authorisation mechanisms. For details take a look at the
; documentation.
;authorisationmodule="CoreAuthorisationModSQLite"

;

; Sets the size of the controls in unlocked (edit) mode of the frontend. This
; defaults to a value of 10 which makes each control be sized to 10px * 10px.
;controls_size=10

;

; Dateformat of the time/dates shown in nagvis (For valid format see PHP docs)
;dateformat="Y-m-d H:i:s"

```



```

;
; File group and mode are applied to all files which are written by NagVis.
; Usually these values can be left as they are. In some rare cases you might
; want to change these values to make the files writeable/readable by some other
; users in a group.
;file_group=""
;file_mode="660"
...

```

观察上述配置文件, 你会注意到绝大多数的参数已经以行首加引号的形式注释掉了, 这说明了 NagVis 采取了自我注释的配置文件, 很多参数都已经配置成默认的了。如果想使默认的参数生效, 你所需要做的仅仅是删除行首的引号, 使默认的参数真正生效, 如图 13-7 所示。

```

1 <?php return 1; ?>
2 the line above is to prevent
3 viewing this file from web.
4 DON'T REMOVE IT!
5
6 -----
7 Default NagVis Configuration File
8 At delivery everything here is commented out. The default values are set in the NagVis code.
9 You can make your changes here, they'll overwrite the default settings.
10 -----
11
12 !!! The sections/variables with a leading ":" won't be recognised by NagVis (commented out) !!!
13 -----
14
15 General options which affect the whole NagVis installation
16 [global]
17 Enable/Disable logging of security related user actions in NagVis. For
18 example user logins and logouts are logged in var/nagvis-audit.log
19 audit_log="1"
20
21 Defines the authentication module to use. By default NagVis uses the built-in
22 SQLite authentication module. On delivery there is no other authentication
23 module available. It is possible to add own authentication modules for
24 supporting other authorisation mechanisms. For details take a look at the
25 documentation.
26 authmodule="CoreAuthModSQLite"
27
28 Defines the authorisation module to use. By default NagVis uses the built-in
29 SQLite authorisation module. On delivery there is no other authorisation
30 module available. It is possible to add own authorisation modules for
31 supporting other authorisation mechanisms. For details take a look at the
32 documentation.
33 authorisationmodule="CoreAuthorisationModSQLite"
34 "nagvis.ini.php" 512L, 18522C

```

图 13-7 自注释的 nagvis.ini.php 文件意味着可以自由编辑 (记得修改前备份)

在 NagVis 安装完毕之后, 除了后台数据源需要特别配置 (在 13.4.2 小节中讲述) 之外, 其他参数采用默认设置, 足以能够支持 NagVis 的正常运行, 接下来的内容主要集中在 nagvis.ini.php 文件的一些重要配置项的相关解释和说明上。

13.4.1 配置 NagVis 的默认参数

使用 Linux 操作系统的 VI 文本编辑工具打开位于 /usr/local/nagvis/etc/ 目录下的 nagvis.ini.php 文件, 可以直接进行编辑, 重启 httpd 服务即可生效。还可以通过 admin 用户登录 NagVis 的 Web 用户界面, 选择菜单 Options→General Configuration 进行配置文件的编辑, 该修改为即时生效, 如图 13-8 所示。

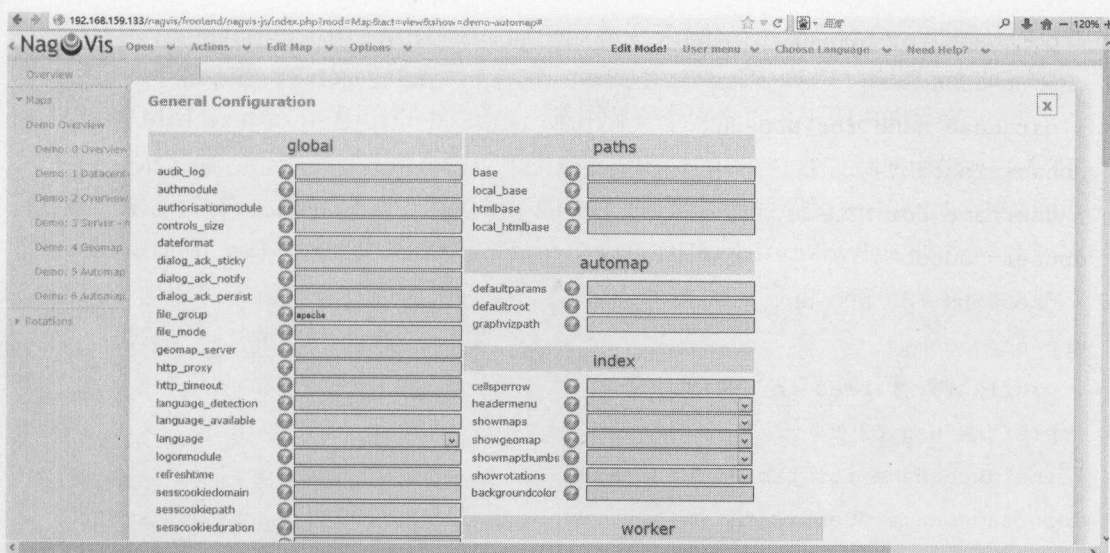


图 13-8 在图形界面设置 NagVis 的参数

使用 VI 打开 `nagvis.ini.php` 文件，定位至 `defaults` 选项组。该选项组指定了地图配置的默认值，后续创建的所有地图对象都可以直接继承该选项组中的默认属性，而不必在修改地图属性时做重复配置。最好是在这里定义大多数地图对象都相同的配置，例如地图中图标的大小、地图自动刷新时间、显示风格、地图上各类监控对象对应的外部超链接等等属性。

而 `defaults` 选项组最关键的属性为 `backend`，即所有地图都采用的默认后台数据源，其配置如下所示：

```
[defaults]
; default backend (id of the default backend)
backend="ndomy_1"
```

`Backend` 属性指定了 NagVis 系统使用哪一类默认的后台，既可以是 `NDOUtils` 数据库，还可以是 `MKLiveStatus` 等 `Socket` 服务。后台的名称可以是任意值，但后台本身仍然需要单独定义，请参考 13.4.2 小节。如果只是想尝试一下，最好保留默认名称 `ndomy_1`。

13.4.2 配置 NagVis 的后台数据源

接下来，我们讨论 NagVis 配置文件中的一个重要配置项——后台数据源，在 NagVis 的配置文件中称为 `backend`（后台数据源）。以下是以使用 `NDOUtils` 采集入库的 `MySQL` 数据库作为后台数据源的一个配置样例：

```
[backend_ndomy_1]
; type of backend - MUST be set
backendtype="ndomy"
;statushost=""
; hostname for NDO-db
dbhost="localhost"
```



```

; portname for NDO-db
dbport=3306
; database name for NDO-db
dbname="ndodb"
; username for NDO-db
dbuser="ndodb"
; password for NDO-db
dbpass="ndodb"
; prefix for tables in NDO-db
dbprefix="nagios_"
; instance name for tables in NDO-db
dbinstancename="Central"
; maximum delay of the NDO Database in seconds
maxtimewithoutupdate=180

```

[backend_ndomy_1]选项组的名称必须包含 defaults 选项组下 backend 参数指定的名称,并遵循如下命名模式: backend_数据源名称_序号,这里默认是 ndomy_1。如果 defaults 选项组下 backend 参数值不与定义的后台数据源部分的任何一个配置匹配,NagVis 将无法正常工作。

backendtype 定义了后台类型,而这里的 ndomy 指的是基于 MySQL 的 NDO 数据库,是唯一能够指定的值。

dbhost 和 dbport 指定了 MySQL 后台主机名或 IP 地址,以及附加的用于访问数据库的 TCP 端口,端口号默认是 3306。dbname 包含了 NDO 数据库的名称,dbuser 和 dbpass 指定访问的用户名和密码。dbprefix 和 dbinstancename 采取 NDOUtils 标准安装所设置的默认值。以上参数涉及到 NDOUtils 组件的相关配置,在位于系统目录 /usr/local/nagios/etc/ 下的 ndomod.cfg 和 ndo2db.cfg 两个文件中可以找到相关的值。一般来说,dbprefix 的值默认为 nagios_,而 dbinstancename 的值为 Central。

需要特别注意的一个参数是 maxtimewithoutupdate。它定义了 Nagios 的状态数据更新最大超时时间(秒)。如果超出了这里指定的最大超时时间,NagVis 会认为后台 MySQL 数据已经过时并显示一个错误。如果 NagVis 连接位于多个 MySQL 服务器上的分布式 NDO 数据库,非常重要的一点是要保证服务器之间的时间是同步的,否则在 NagVis 遇到时间差大于 maxtimewithoutupdate 秒时,将无法正常工作。

在 NagVis 的早期版本中,MySQL 数据库是其唯一的数据源。通过配置 MySQL 数据库的服务 IP、端口号、用户名和密码等信息,NagVis 可以访问 Nagios 的后台数据库,从库中读取 Nagios 收集到的各类监控数据,并将其展示出来。从 NagVis 的 1.5 版本开始,引入了 MKLivestatus 组件作为默认的后台数据源。相对于 MySQL 数据库而言,MKLivestatus 是一种全新的 Nagios 事件传播组件。简而言之,MKLivestatus 可以对外提供直接的 socket 服务,从而使程序可以绕开数据库,直接访问 Nagios 采集到的各类信息。与 MySQL 相比,MKLivestatus 具备易安装、易管理,以及实时快速对外提供数据等多种特性,同时不需要数据库管理的专业知识。

在 NagVis 的近期版本中, MKLivestatus 已经取代了 NDO, 成为 NagVis 的默认后台数据源, 但这并不意味着 NDO 已经遭到了抛弃。事实上, NagVis 的默认配置文件同时提供了 MySQL 和 MKLivestatus 作为可选的数据源, 采用哪一个完全取决于用户的爱好。

采用 MKLivestatus 作为默认后台数据源的相关设置, 其中 defaults 选项组中的 backend 选项值应该是 live_1, 与 backend_live_1 选项组中的 live_1 一致。而后续的 backendtype 选项的值设置为 mklivestatus, socket 设置为系统默认的 unix:/usr/local/nagios/var/rw/live 即可。

```
[defaults]
; default backend (id of the default backend)
backend="live_1"
[backend_live_1]
backendtype="mklivestatus"
socket="unix:/usr/local/nagios/var/rw/live"
```

一旦 NagVis 出现问题, 在监控大屏上会出现相应的报错信息, 此时需要查看操作系统的日志文件 /var/log/messages。该文件记录了 NagVis 工作时产生的各类日志信息, 通常错误是无法连接 MySQL 数据库, 此时你会看到诸如 “Could not open data sink!” 的报错信息, 此时就需要检查 MySQL 数据库是否正常工作、NagVis 后台数据源等等配置项是否正确、用户是否具备权限读写相应目录或者文件等等, 必要时可以求助于数据库管理员。

13.5 NagVis 地图介绍

如果正确安装了 NagVis, 那么在登录之后的页面中, 你将会注意到 NagVis 已经默认显示了一些名称以 Demo 开头的地图, 如图 13-9 所示, 是 NagVis 提供的自动拓扑图样例:

一看到 NagVis 的运行界面, 也许你已经迫不及待地想要配置并显示自己的第一张监控地图了吧, 但我们还要等一等。在配置监控地图之前, 一定要先了解配置监控地图的关键组件——背景图片。

NagVis 中的背景图片是泛指在监控屏幕上, 与人们看到的监控图标相对应的、起到衬托作用或者展示作用的独立图像。它是一张独立的图片, 格式一般为 PNG, 与放置其上的主机监控图标或者服务监控图标没有相互的关联。简而言之, 背景图片能够起到流程展示、位置辨识、大屏幕装饰等作用, 是值班员、尤其是更高一级的 IT 管理人员理解 IT 运维监控流程和系统架构不可或缺的帮手。

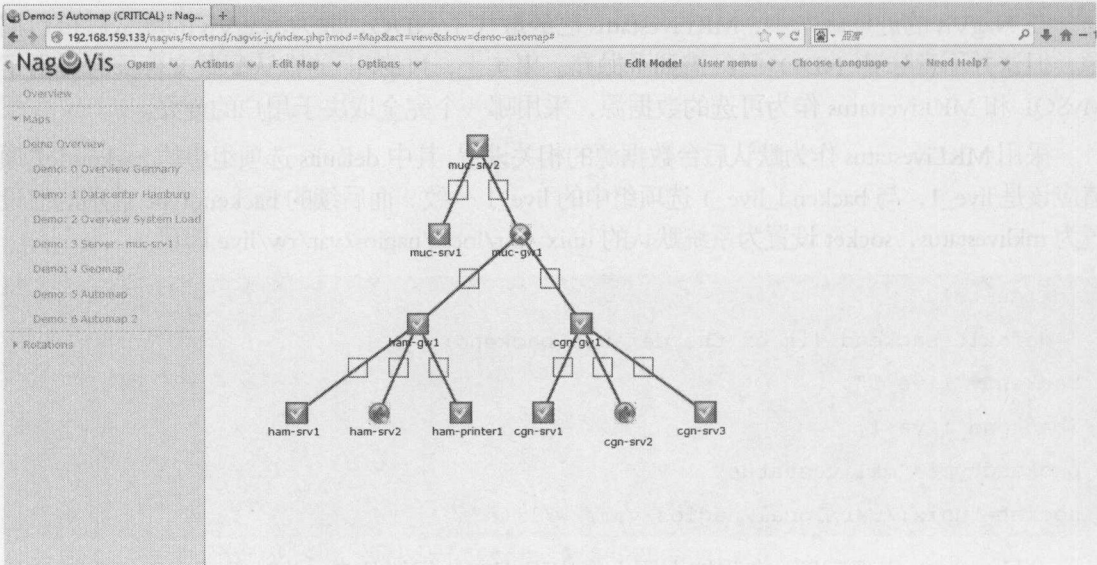


图 13-9 NagVis 提供的拓扑图样例

如图 13-10 所示，将机柜照片作为背景图片引入 NagVis 中，当某一部件出现问题后，可迅速在图片上定位问题所在。



图 13-10 将照片作为 NagVis 的地图背景图片

13.6 NagVis 地图的配置管理

打开浏览器，使用 admin 用户登录 NagVis 的 Web 界面，选择 NagVis 界面上方的菜单 Options→Manage Maps，可以进入地图管理页面，如图 13-11 所示。

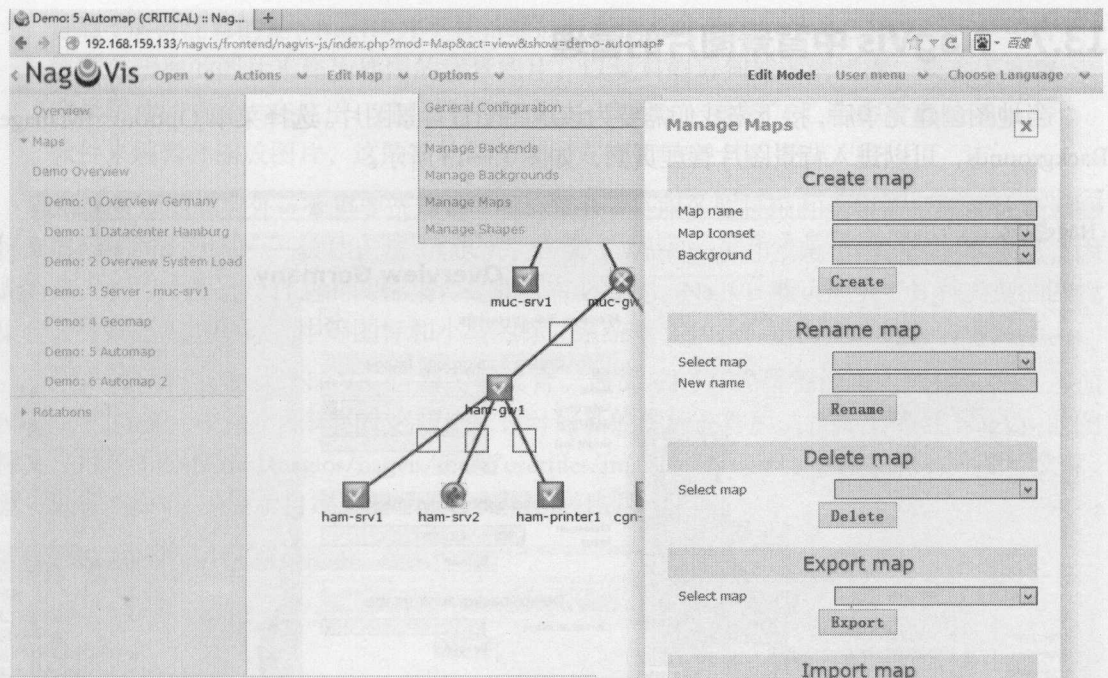


图 13-11 管理 NagVis 中的地图

如上图 13-12 所示，在地图管理页面中，可以对地图进行创建、重命名、删除、导入导出等工作。例如，在创建地图选项中，只需输入欲创建的地图名，单击 Create 按钮即可按照默认格式创建一张 NagVis 地图。此后在浏览器左侧的 OverView 列表中将会看到刚刚创建的地图。由于此时地图上尚未摆放任何图片，因此看起来是一张空白地图，还可以在上面导入背景图片和各种类型的监控图标，着手一系列的设计工作，方能正式向用户展示该地图，如图 13-12 所示。

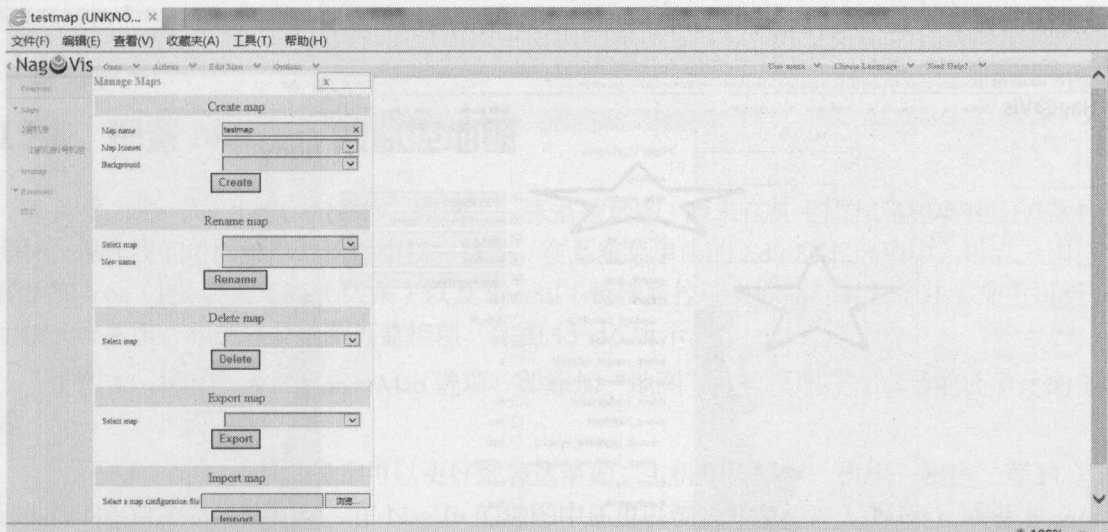


图 13-12 地图管理页面

13.7 NagVis 中背景图片的管理

当地图创建完毕后，接下来我们需要为该地图配置背景图片。选择菜单 Options→Manage Backgrounds，可以进入背景图片管理页面，如图 13-13 所示。



图 13-13 NagVis 中的背景图片管理

上图 13-13 中的 Upload background image（上传背景图片）选项允许选择一张合适分辨率的、格式为 PNG 的图片上传到 NagVis 中。用户可以上传多张背景图片，以便后续配置地图选项时可以根据监控业务需要来选择合适的图片。

接下来在左侧的地图列表中选择需要配置该背景图片的地图，选择菜单 Edit Map→Map Options（地图选项），在 map_image（地图背景）下拉列表中选择刚刚上传的背景图片，将该地图的背景设置为之前上传的图片，如图 13-14 所示。

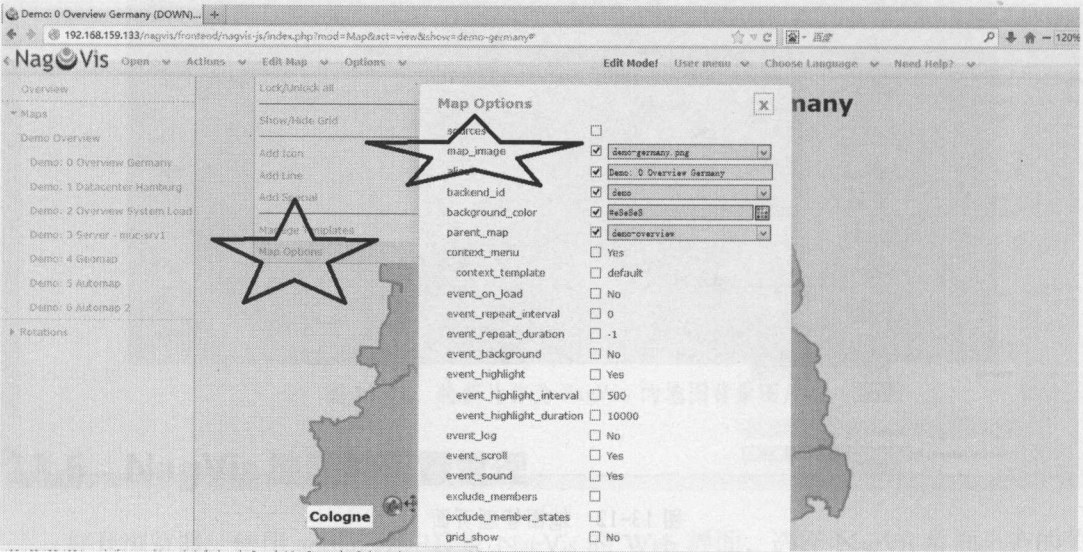
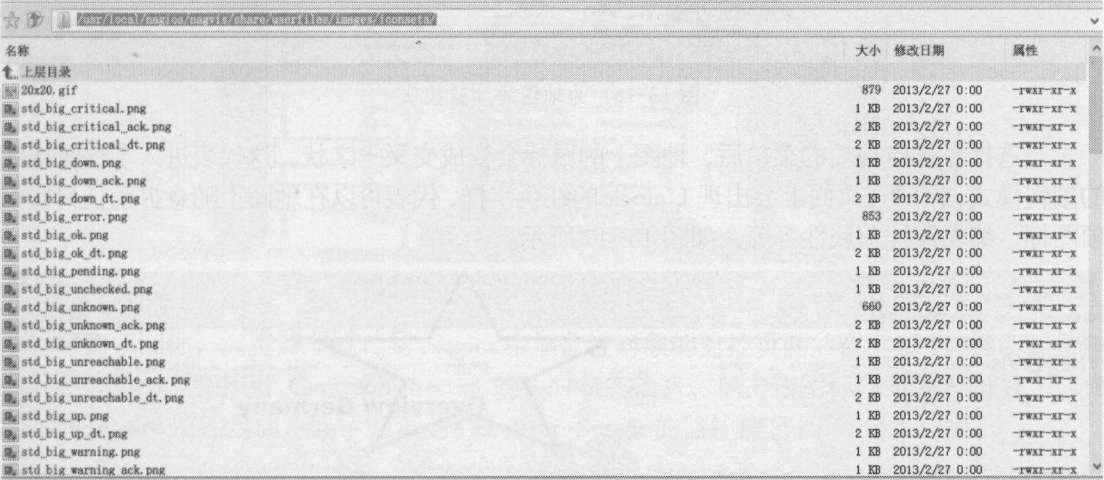


图 13-14 为 NagVis 的地图配置背景图片

注意：值得注意的是，Nagvis 并不具备缩放背景图片的功能，因此必须选择一张适合屏幕分辨率的图片来作为地图的背景图片，以免只能看到地图的局部，失去了布置背景图片的意义。在这里，建议选择 Windows 自带的“画图”工具或者 Photoshop 等软件来编辑并缩放图片，这两种软件都可以方便地导出 PNG 格式的图片。

NagVis 地图的另外一个重要选项是“图标集合”。服务监控项和主机监控项都依赖于图标来展示不同的状态——例如正常（OK）、告警（Warning）和紧急（Critical）等状态，因此为每张监控地图选择合适的图标集合是非常有必要的。NagVis 默认提供了 3 种类型的图标集合，分别是大型图标、中等图标和小型图标，系统默认会选择这套图标进行告警展示。

用户可以自己设计 NagVis 告警图标集合。最简单的方式是备份原有图标文件，遵循 NagVis 图标集合中每个图标的命名规则，设计好新的图标文件后，直接上传至 NagVis 的图标文件目录（/usr/local/nagios/nagvis/share/userfiles/images/iconsets/），覆盖原有的图标文件，就可以在 NagVis 中展示自定义图标了，如图 13-15 所示。



名称	大小	修改日期	属性
⬆ 上一步目录			
20x20.gif	879	2013/2/27 0:00	-TWEI-XI-X
std_big_critical.png	1 KB	2013/2/27 0:00	-TWEI-XI-X
std_big_critical_ack.png	2 KB	2013/2/27 0:00	-TWEI-XI-X
std_big_critical_dt.png	2 KB	2013/2/27 0:00	-TWEI-XI-X
std_big_down.png	1 KB	2013/2/27 0:00	-TWEI-XI-X
std_big_down_ack.png	1 KB	2013/2/27 0:00	-TWEI-XI-X
std_big_down_dt.png	2 KB	2013/2/27 0:00	-TWEI-XI-X
std_big_error.png	853	2013/2/27 0:00	-TWEI-XI-X
std_big_ok.png	1 KB	2013/2/27 0:00	-TWEI-XI-X
std_big_ok_dt.png	2 KB	2013/2/27 0:00	-TWEI-XI-X
std_big_pending.png	1 KB	2013/2/27 0:00	-TWEI-XI-X
std_big_unchecked.png	1 KB	2013/2/27 0:00	-TWEI-XI-X
std_big_unknown.png	660	2013/2/27 0:00	-TWEI-XI-X
std_big_unknown_ack.png	2 KB	2013/2/27 0:00	-TWEI-XI-X
std_big_unknown_dt.png	2 KB	2013/2/27 0:00	-TWEI-XI-X
std_big_unreachable.png	1 KB	2013/2/27 0:00	-TWEI-XI-X
std_big_unreachable_ack.png	1 KB	2013/2/27 0:00	-TWEI-XI-X
std_big_unreachable_dt.png	2 KB	2013/2/27 0:00	-TWEI-XI-X
std_big_up.png	1 KB	2013/2/27 0:00	-TWEI-XI-X
std_big_up_dt.png	2 KB	2013/2/27 0:00	-TWEI-XI-X
std_big_warning.png	1 KB	2013/2/27 0:00	-TWEI-XI-X
std_big_warning_ack.png	1 KB	2013/2/27 0:00	-TWEI-XI-X

图 13-15 NagVis 中的图标文件

13.8 配置 NagVis 的监控地图

到此为止，我们仅仅为 NagVis 新增了一张监控地图，尚未在其上添加监控项并配置监控图标，现在我们可以着手做这件事情。首先，在系统菜单中的 Edit Map 选项中，可以分别选择添加 Icon（图标）、Line（线条）以及 Special（特殊监控项）等监控对象，并添加主机组、主机、服务组、服务以及地图等监控项，如图 13-16 所示。

上图 13-16 中，Add Icon→Map 选项，即添加“地图”监控项即在本章节前面所述的创建“父—子”地图关系。

“父—子”地图关系的设定可以更好地表达诸如“主机—主机组—机柜—机房”等 IT 基础设施的层级关系，同样地，在 NagVis 的地图中还可以创建一个独立于 NagVis 或者 Nagios 的外部服务或者外部主机监控项，比如，如果想监控一个外部网站的健康状况，就可以创建一个代表外部 URL 链接的监控项，借助于该监控项的图标，就可以在 NagVis 的地图上显示

其他网段的网站服务状态。

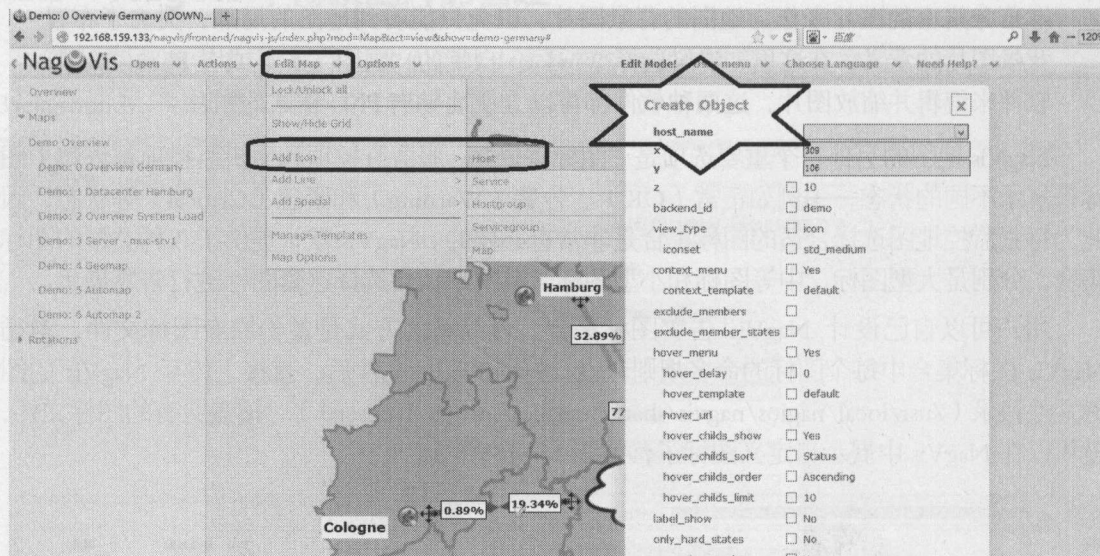


图 13-16 为地图添加监控项

当选择添加监控项的菜单后，地图上的鼠标会变成交叉十字状，这代表进入了这张地图的编辑模式，同时在页面上会出现 Unlock 的红色字样，代表可以在地图上随意挪动任何监控项图标、编辑监控项属性等等，如图 13-17 所示。

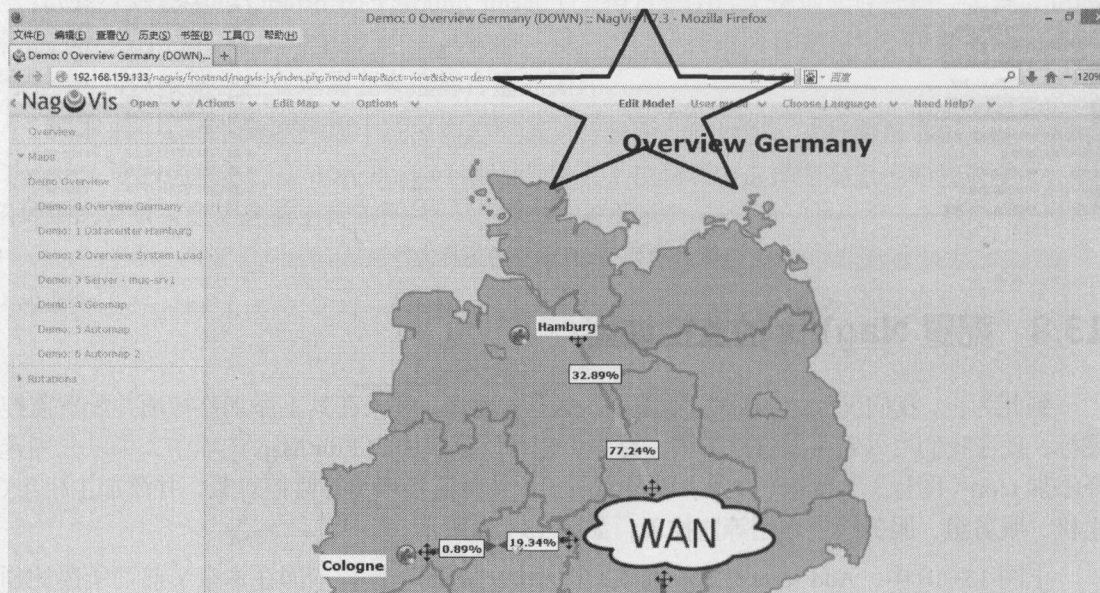


图 13-17 NagVis 的编辑模式

完成监控对象图标的摆放和编辑后，不要忘记选择菜单 Edit Map→Lock/Unlock all，退出编辑状态，然后该地图就会实时显示最后的完成视图，并开始实时显示监控对象的实际状态。如果再想对该地图进行编辑，再次选择该菜单项即可。

当代表具体监控对象的监控图标、连线被摆放在如上图 13-17 所示的背景地图之后，就共同

组合成了代表具体业务逻辑的拓扑图，也就可以被投放到监控大屏幕上上进行实时监控展示了。

13.9 设置 NagVis 图标的超链接

在 NagVis 中，每个监控图标或者连线都代表 Centreon 和 Nagios 中的一个或者一组监控对象。当地图编辑完毕，解除锁定状态后，鼠标悬停在地图的某个监控图标或者连线上，即弹出该监控图标的当前状态菜单，同时显示出该图标为超链接，如图 13-18 所示。

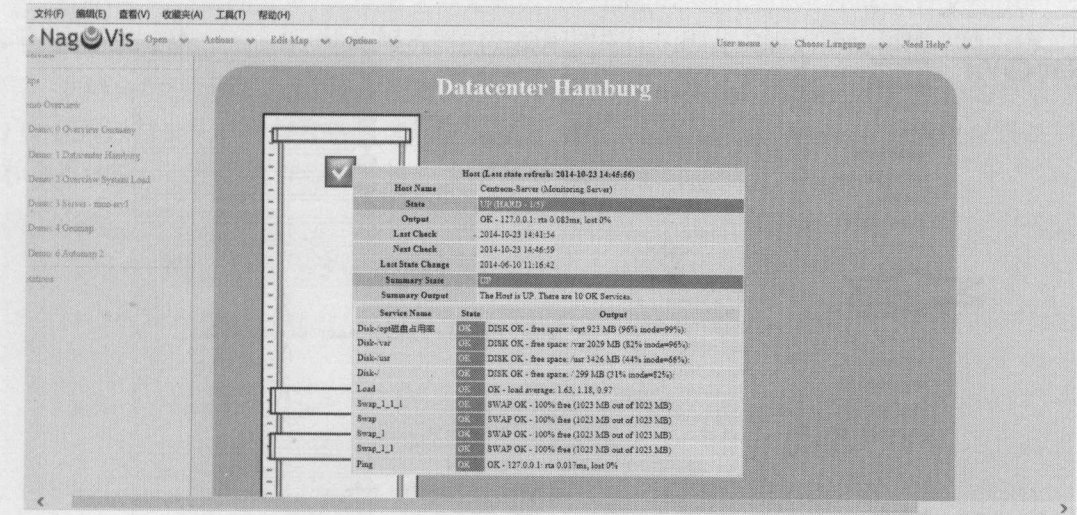


图 13-18 NagVis 中监控图标的状态

单击该图标，浏览器即跳转到 Nagios 中该监控对象的监控页面。在上图 13-18 中，绿色监控图标显示了监控机 Centreon-Server 的实时状态信息，单击该图标，即进入如图 13-19 所示的 Nagios 监控页面，显示的正是 Centreon-Server 的监控信息。

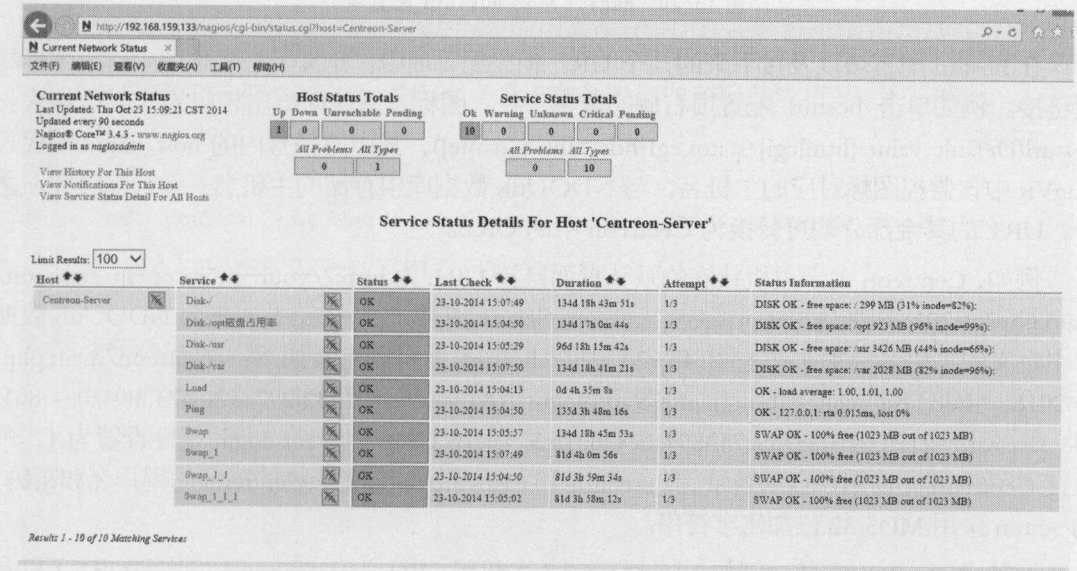


图 13-19 NagVis 到 Nagios 的链接

在浏览器中单击监控图标，除了能够默认跳转到 Nagios 相应的监控对象 Web 页面中外，还能够调整为跳转到 Centreon 的 Web 页面，并能够直接定位到相应的监控对象，遵循下列步骤即可实现：

在 NagVis 中选择菜单 Options→General Configuration，进入通用选项配置页面。在弹出的页面中定位 hosturl 配置项，如图 13-20 所示。

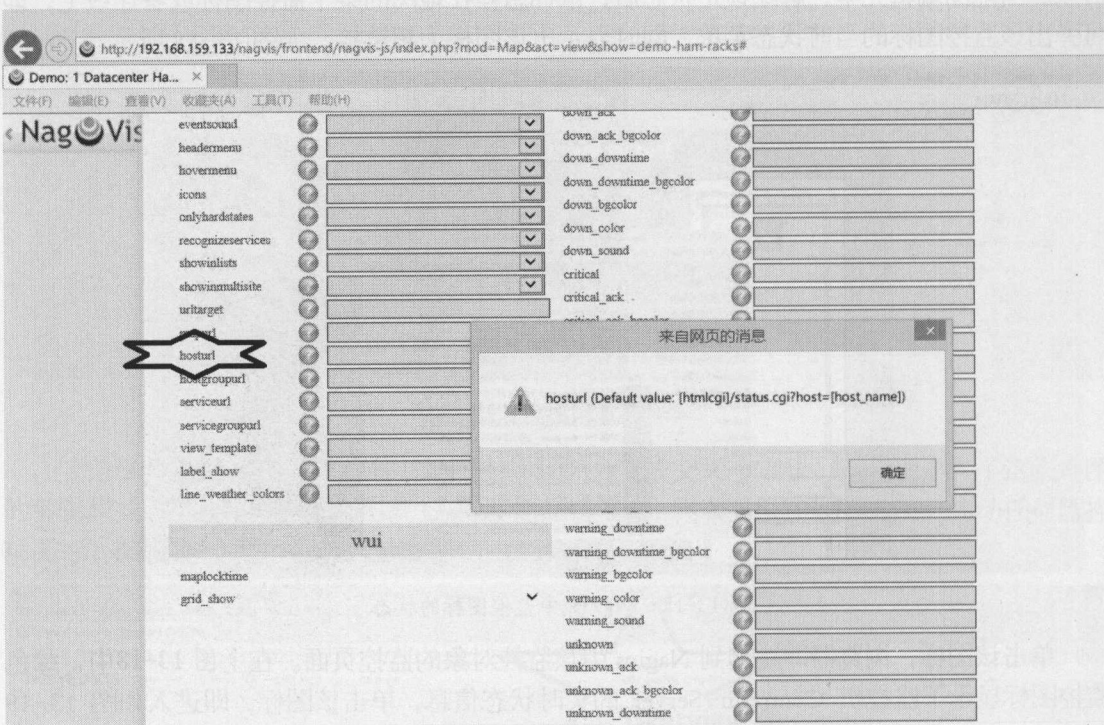


图 13-20 NagVis 中的 hosturl 配置项

在 hosturl 配置项以及接下来的几个 URL 相关配置项中，定义了每张地图中监控图标的超链接。例如单击 hosturl 配置项右侧的蓝色“？”图标，系统即给出相应的 URL 格式：hosturl(Default value:[htmlcgi]/status.cgi?host=[host_name])，其中方括号内的 host_name 即代表 NagVis 中该监控图标对应的主机名，与 NDOUtils 数据库中存储的主机名一致，而除此项之外，URL 的其余部分均可替换为 Centreon 中的 URL。

例如，Centreon 关于主机状态的默认页面链接 URL 为 http://your-monitor-ip/centreon/main.php?p=201&o=hd&host_name=Centreon-Server，其中 Centreon-Server 为 NDOUtils 数据库中定义的主机名，那么在上图 13-20 中的 hosturl 就可以定义如下：/centreon/main.php?p=201&o=hd&host_name=[host_name]&autologin=1&useralias=21232f297a57a5a743894a0e4a801fc3&password=21232f297a57a5a743894a0e4a801fc3。上述 URL 中，autologin 属性设置为 1，可以使 Centreon 允许用户自动登录，而 useralias 和 password 即为登录 Centreon 的用户名和密码，为 admin 采用 MD5 32 位加密字符串。

而其余的 URL 定义，例如 hostgroupurl（主机组 URL），serviceurl（服务 URL）以及 servicegroupurl（服务组 URL）等等，都可以根据 Centreon 中的相关 URL 格式进行定义。

13.10 设置 NagVis 的 Web 界面为自动登录

在实际的监控业务中，NagVis 中的地图常常投放到监控中心的大屏上，供一线监控人员监控，而对于一线人员来说，只要求他们能够检测并响应监控对象的告警即可，并不要求他们掌握登录 NagVis 的用户名和密码，因此有必要使 NagVis 的 Web 用户界面能够自动登录，并设置为浏览器的默认主页，使一线人员一启动浏览器就能看到监控视图。

NagVis 支持自动登录，其设置主要是修改 `/etc/httpd/conf.d/nagvis.conf` 配置文件中的某些项。由于该文件为 Apache Web 服务器的重要配置文件，因此编辑前务必保留一份备份。

支持自动登录的 `nagvis.conf` 配置文件如下所示，其中粗体的 `admin` 部分为自动登录的用户，还可以替换成其他仅具备只读功能的用户，确保一线人员对 Centreon 系统仅仅具备只读和查看功能，取消其管理权限，进一步增强平台的安全性。

```
# NagVis Apache2 sample configuration file
#
# #####

Alias /nagvis "/usr/local/nagvis/share"

<Directory "/usr/local/nagvis/share">
    Options FollowSymLinks
    AllowOverride None
    Order allow,deny
    Allow from all

# To enable Nagios basic auth on NagVis use the following options
# Just uncomment it. Maybe you need to adjust the path to the
# Auth user file.
#
# If you use the NagVis internal auth mechanism based on the web
# for you won't need this.
#
#AuthName "NagVis Access"
#AuthType Basic
#AuthUserFile /usr/local/nagios/etc/htpasswd.users
#Require valid-user

# With installed and enabled mod_rewrite there are several redirections
# available to fix deprecated and/or wrong urls. None of those rules is
```



```
# mandatory to get NagVis working.
<IfModule mod_rewrite.c>
    RewriteEngine On
    RewriteBase /nagvis

    # Use mod_rewrite for old url redirection even if there are php files
    which
    # redirect the queries itselfs. In some cases the mod_rewrite redirect
    # is better than the php redirect.
    #
    # Using the php redirect seems to be better in some cases where https/http
    servers
    # are mixed. For example in OMD setups where using apache own mode and
    https in the
    # frontend and http in the backend apache servers.
    #
    # Disabling this redirect by default in the hope that the php direct works
    better.
    #RewriteCond %{REQUEST_URI}
    ^/nagvis(/config\.php|/index\.php|/)(\?.*|)$
    #RewriteRule ^(.*)$ /nagvis/frontend/nagvis-js/%1%2 [R=301,L]

    # Redirect old regular map links
    RewriteCond %{REQUEST_URI} ^/nagvis/frontend/(wui|nagvis-js)
    RewriteCond %{QUERY_STRING} map=(.*)
    RewriteRule ^(.*)$
    /nagvis/frontend/nagvis-js/index.php?mod=Map&act=view&show=%1
    [R=301,L]

    # Without map= param
    RewriteCond %{REQUEST_URI} ^/nagvis/frontend(/wui)?/?(index.php)?$
    RewriteRule ^(.*)$ /nagvis/frontend/nagvis-js/index.php [R=301,L]

    # Redirect old rotation calls
    RewriteCond %{REQUEST_URI} ^/nagvis/frontend/nagvis-js
    RewriteCond %{QUERY_STRING} !mod
    RewriteCond %{QUERY_STRING} rotation=(.*)
    RewriteRule ^(.*)$
    /nagvis/frontend/nagvis-js/index.php?mod=Rotation&act=view&show=%1
    [R=301,L]
</IfModule>
```

```
</Directory>
RewriteEngine on
RewriteLock var/log/rewrite.lock
RewriteLog /dev/null
RewriteLogLevel 0

# The following line is the really important step,
# it tells the webserver that the user "nagiosadmin" has
# successfully authenticated and is sending the request,
# regardless who is sending it really.

RewriteRule /nagvis/ - [E=REMOTE_USER:admin]
```




读书笔记

Handwriting practice area with horizontal lines.

第 14 章

构建企业级 IT 运维监控系统

随着信息技术的发展，IT 对社会的影响日渐加深，人们对 IT 的态度也不再是“我能为 IT 做什么”，而是转变为“IT 能为我做什么”。IT 不可避免地变成一种服务，IT 业也成为服务业中的一员。正如前任 SUN 公司 CEO 麦克尼利预测的那样：“将来软件业将不再存在，也不应该存在。所有的事情就是服务，而没有产品。人们编写软件，这是肯定的，但他们在创造服务，而非产品。”

14.1 IT 服务管理和 ITIL

什么是 IT 服务

IT 服务就是由 IT 服务提供商提供的，综合利用人、资源和程序以让客户感觉协调一致的方式，满足客户的一种或多种的信息需求。换句话说，IT 服务提供商需要深入理解客户和他们的业务，一方面正确认识客户的业务模型，一方面要了解客户的实际需求，然后通过高质量的 IT 服务为客户创造出更多的价值，以提高 IT 投资的回报。

服务也是一种产品，可以被开发、制造、交付、销售和消费。但与物质产品相比具有无形性，因而又具有其自身的特性：服务因人员的不同、事件的变化会出现差异，从而导致服务的构成和质量水准难以固定，具有差异性；服务作为一系列的活动或过程，其生产和消费同时进行，两者在时间上不可分离，因而具有不可分离性。这些特征决定了在管理上我们很难用传统的管理思想和方法来管理 IT 服务，而必须研究“IT 服务管理”的手段和方法，以便更好地运用 IT 技术更好地解决问题。

而 ITIL 正是业界普遍采用的一系列 IT 服务管理的实际标准及最佳实践指南，目前已经成为业界通用的事实标准。它以流程为导向、以客户为中心，通过整合 IT 服务与企业服务，提高企业的 IT 服务提供和服务支持的能力和水平。ITIL 可以引导组织高效和有效地使用技术，让既有的信息化资源发挥更大的效能。

什么是 ITIL

根据百度百科相关词条，ITIL 即 IT 基础架构库(Information Technology Infrastructure Library, ITIL, 信息技术基础架构库)由英国政府部门 CCTA(Central Computing and Telecommunications Agency)在 20 世纪 80 年代末制订，现由英国商务部 OGC(Office of Government Commerce)负责管理，主要适用于 IT 服务管理(ITSM)。ITIL 为企业的 IT 服务管理实践提供了一个客观、严谨、可量化的标准和规范。

14.2 IT 运维监控系统与 ITIL 的关系

14.2.1 ITIL 的产生与发展

早在 20 世纪 80 年代中期，英国政府部门发现提供给他们的 IT 服务质量不佳，于是要求当时的政府计算机和电信管理局启动一个项目对此进行调查，并开发一套有效的，以及可进行财务计量的 IT 资源使用和管理方法以供本国的政府部门和私有部门使用。同时，这种方法还应该是独立于厂商的并且可适用于不同规模、不同技术和业务需求的组织。这个项目最终成果是一套公开出版的 IT 管理指南——ITIL。

虽然 ITIL 最早是为英国政府开发的，但是在 20 世纪 90 年代初期，它很快就在欧洲其他国家和地区流行起来，继而成为事实上的欧洲 IT 服务管理标准。随后，ITIL 又被引入到美国、南非和澳大利亚等国家。从 2000 年开始，ITIL 的管理方英国商务部(Office of Government Commerce, OGC)又组织有关力量对 ITIL 进行了较大的扩充和完善，最终逐渐形成了 ITIL V2 的完整知识体系。

2001 年英国标准协会 (British Standard Institute, BSI) 在国际 IT 服务管理论坛年会上正式发布了以 ITIL 为基础的 IT 服务管理英国国家标准 BS15000。2005 年 12 月, BS15000 正式发布为国际标准 ISO20000。2007 年 5 月 30 日, OGC 在全球发布了 ITIL 最新版, 即 ITIL V3。

ITIL 是业界普遍采用的一系列 IT 服务管理的实际标准及最佳实践指南, 目前已经成为业界通用的事实标准。它以流程为导向、以客户为中心, 通过整合 IT 服务与企业服务, 提高企业的 IT 服务提供和服务支持的能力和水平。ITIL 可以引导组织高效和有效地使用技术, 让既有的信息化资源发挥更大的效能。

14.2.2 ITIL 的管理框架简介

ITIL 在各大企业中运用较多的是 V2 版本, V3 版本是近几年逐渐成熟起来的, 是在 V2 的基础上进行的扩充和完善, 下面对 V2 和 V3 的框架进行简要的介绍。

ITIL V2

ITIL V2 具体包括以下 6 方面内容:

- 业务管理: 在提供 IT 服务的时候, 首先应该考虑业务需求, 根据业务需求来确定 IT 需求; 业务管理模块指导管理者以自己习惯的思维模式分析 IT 问题, 了解 IT 基础架构支持业务流程的能力, 以及 IT 服务管理在提供端到端 IT 服务过程中的作用。
- IT 基础架构管理: 侧重于从技术角度对基础设施进行管理。覆盖了 IT 基础设施管理的所有方面, 包括识别业务需求、实施和部署、对基础设施进行支持和维护等方面。目标是确保 IT 基础设施架构稳定可靠, 能够满足业务需求和支撑业务运作。
- 应用管理: 为了确保应用系统满足客户需求并方便对其进行支持和维护, IT 服务管理的职能应该合理地延伸, 介入应用系统的开发、测试和部署。应用管理模块指导 IT 服务提供方协调应用系统的开发和维护, 以使它们一致地为客户的业务运作提供支持和服
- 安全管理: 安全管理是 1999 年新增到 ITIL 中的模块。目标是保护 IT 基础架构, 使其避免未经授权的使用, 从确定安全需求、策略和方法的角度指导如何进行安全管理。
- IT 服务规划与实施: 其作用是指导如何实施上述模块中的各个流程, 包括对这些流程的整合。指导客户确立远景目标, 分析和评价现状, 确定合理的目标并进行差距分析, 确定任务的优先级, 以及对流程的实施情况进行评审。
- 服务管理: 服务管理模块是 ITIL V2 的核心模块。具体又可以分为服务交付和服务支持两个模块, 包括十大流程和一项服务台职能, 具体如表 14-1 所示:

表 14-1 ITIL 服务交付和服务支持

服务交付	服务级别管理
	可用性管理
	IT 服务财务管理
	IT 服务持续性管理
	能力管理

续表

服务支持	服务台（职能）
	事件管理
	问题管理
	配置管理
	变更管理
	发布管理

表 14-1 中的服务交付主要面向服务付费的机构和个人客户。它的任务是根据组织的业务需求，对服务能力、持续性、可用性等服务级别目标进行规划和设计，同时还必须考虑到实现这些服务目标所需要耗费的成本。也就是说，在进行服务提供流程设计时，必须在服务级别目标和服务成本之间进行合理的权衡。由于这些管理流程必须解决“客户需要什么”、“为满足客户需求需要哪些资源”、“这些资源的成本是多少”、“如何在服务成本和服务效益（达到的服务级别）之间选择恰当的平衡点”等问题，因而服务交付所包括的这 5 个核心流程均属于战术层次的服务管理流程，它们之间的关系如图 14-1 所示。

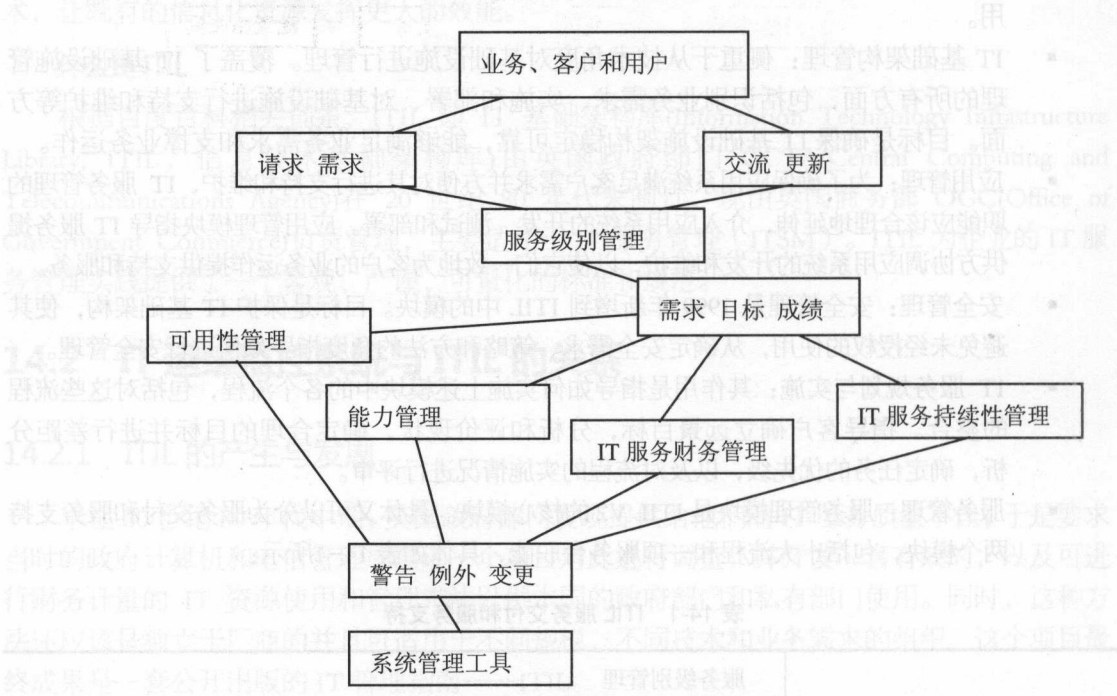


图 14-1 服务交付各流程间的关系

而服务支持主要面向用户（End-Users），用于确保用户得到适当的服务以支持组织的业务功能，确保 IT 服务提供方所提供的服务质量符合服务级别协议的要求。服务级别协议（Service Level Agreement，简称 SLA，参考 7.1.4 小节）规定了 IT 服务提供方应该给客户提供什么程度或者级别的服务，以及没有达到相应级别的服务时，如何进行补偿。服务支持各流

程间的关系如图 14-2 所示。

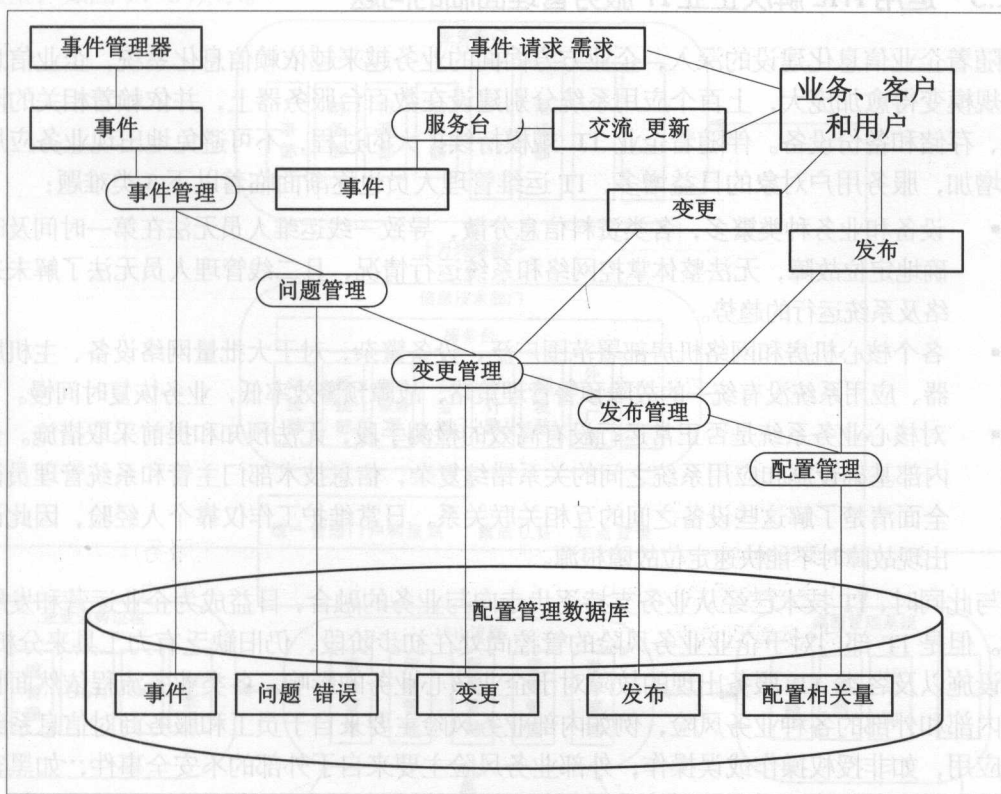


图 14-2 服务支持各流程间的关系

ITIL V3

ITIL V3 是一个巩固和提高 ITIL 最佳实践的过程，也是“当前最佳实践”的精髓。“当前最佳实践”规定了行业实践中的前沿信息，并且会随着客户需求的变化而不断变化。OGC 对 ITIL V2 中的重要内容加以精简，然后将其收录到 ITIL V3 中。ITIL V3 的结构框架和内容来源于大量的公众评议会及行业管理者的意见。同时它也囊括了 V2 中仍被 ITSM 团体广泛实践和运用的那部分内容。

ITIL V3 主要增加了部分新概念，尤其是引入了“生命周期”这个概念。IT 服务从开始到结束的整个过程，就是服务管理的生命周期。当开展一项服务时，组织中不同的管理层和成员都参与到该服务的生命周期中，包括决策、计划、设计、开发、测试、发布、运行和改进等活动。借助于“IT 服务管理生命周期”的贯穿，ITIL V3 将 V2 中的各个流程有机地整合在了一起。但严格说起来，V3 只是 V2 的加强版，它补充并解释了 V2 的不足之处，在前者的基础上增加了一些营销方法与流程，并解释了 ITIL 在不同的行业该如何切入，使得 ITIL 跟企业的关系更加紧密。

IT 服务管理生命周期模型的引入改变了模块之间相互割裂、独立实施的局面，从战略、战术和运作三个层面针对业务和 IT 快速变化提出服务管理实践方法。它通过连贯的逻辑体系，以服务战略作为总纲，通过服务设计、服务转换和服务运作加以实施，并借助持续服务改进不断完善整个过程，使 IT 服务管理的实施过程被有机整合为一个良性循环的整体。

14.2.3 运用 ITIL 解决企业 IT 服务管理面临的问题

随着企业信息化建设的深入，企业方方面面的业务越来越依赖信息化系统，企业信息系统的规模变得愈加庞大，上百个应用系统分别建设在数百台服务器上，并依赖着相关的网络设备、存储和备份设备。伴随着企业 IT 规模持续扩大的过程，不可避免地出现业务应用的不断增加，服务用户对象的日益增多，IT 运维管理人员也逐渐面临着以下 3 类难题：

- 设备和业务种类繁多，各类资料信息分散，导致一线运维人员无法在第一时间及时准确地定位故障，无法整体掌控网络和系统运行情况，且二线管理人员无法了解未来网络及系统运行的趋势。
- 各个核心机房和网络机房部署范围广泛，设备繁杂，对于大批量网络设备、主机服务器、应用系统没有统一的故障预警管理策略，故障预警效率低，业务恢复时间慢。
- 对核心业务系统是否正常运行没有高效的检测手段，无法预知和提前采取措施。企业内部基础设施和应用系统之间的关系错综复杂，信息技术部门主管和系统管理员很难全面清楚了解这些设备之间的互相关联关系，日常维护工作仅靠个人经验，因此设备出现故障时不能快速定位故障根源。

与此同时，IT 技术已经从业务支持逐步走向与业务的融合，日益成为企业运营和发展的支柱。但是 IT 部门对于企业业务风险的管控尚处在初步阶段，仍旧缺乏有力工具来分析 IT 基础设施以及各类 IT 服务出现的故障对于企业核心业务的影响。各类业务流程依然面临着来自内部和外部的各种业务风险，例如内部业务风险主要来自于员工和服务商对信息系统的不当应用，如非授权操作或误操作；外部业务风险主要来自于外部的不安全事件，如黑客攻击、机房环境变化等。对应用系统进行业务监控，能够及时识别业务风险，有效进行相应的主动规避操作，避免造成损失。

多年的运维经验告诉我们，IT 基础设施的故障仅占 IT 系统总故障的 20%，而各类业务系统故障占到 IT 系统总故障的 60%~80%，业务系统的每一个微小的故障都可能会导致业务的重大损失。针对诸如此类问题，国内外 IT 管理实践普遍的办法是从技术入手来解决。从我国各单位 IT 管理系统建设的历程来看，基本也是首先从“技术”角度入手，对 IT 系统进行管理。

企业最早实施的管理 IT 系统的手段就是建设 IT 运维监控管理系统。监控管理相当于信息技术部运维人员的千里眼和顺风耳，利用信息化技术观察基础设施内网络系统、主机、数据库、应用等管理对象的性能、健康状况，收集基础设施运行过程中的警告或故障信息，进行存储、过滤、翻译、分类、关联，按照问题的性质、类别和严重程度分别采取不同的预警处理措施。这样运维人员就可以在监控管理平台上查看系统运行状况，在某种程度上减轻了运维人员的工作量，提高了运维人员的工作效率，达到了预防性维护的目标。

随着 IT 运维监控的实践，ITIL 最佳实践流程也逐渐引起了广大企业的重视。通过建立统一服务台，实现了业务部门与 ITIL 部门的单一联系点，并引入事件管理、问题管理、变更管理、配置管理、发布管理等流程来规范企业内人员的运维活动，形成了一系列的运维规程。通过 ITIL 流程的实施，各企业内运维人员的服务意识逐步增强，运维工作量化逐步实现。

一般来说，企业的 IT 部门在量身定做实施 ITIL 的过程中，可以针对当前企业 IT 管理建设现状，依据 IT 服务管理过程中人员、流程与技术三要素，提出一个整合的 IT 服务管理整

合模型，如图 14-3 所示。

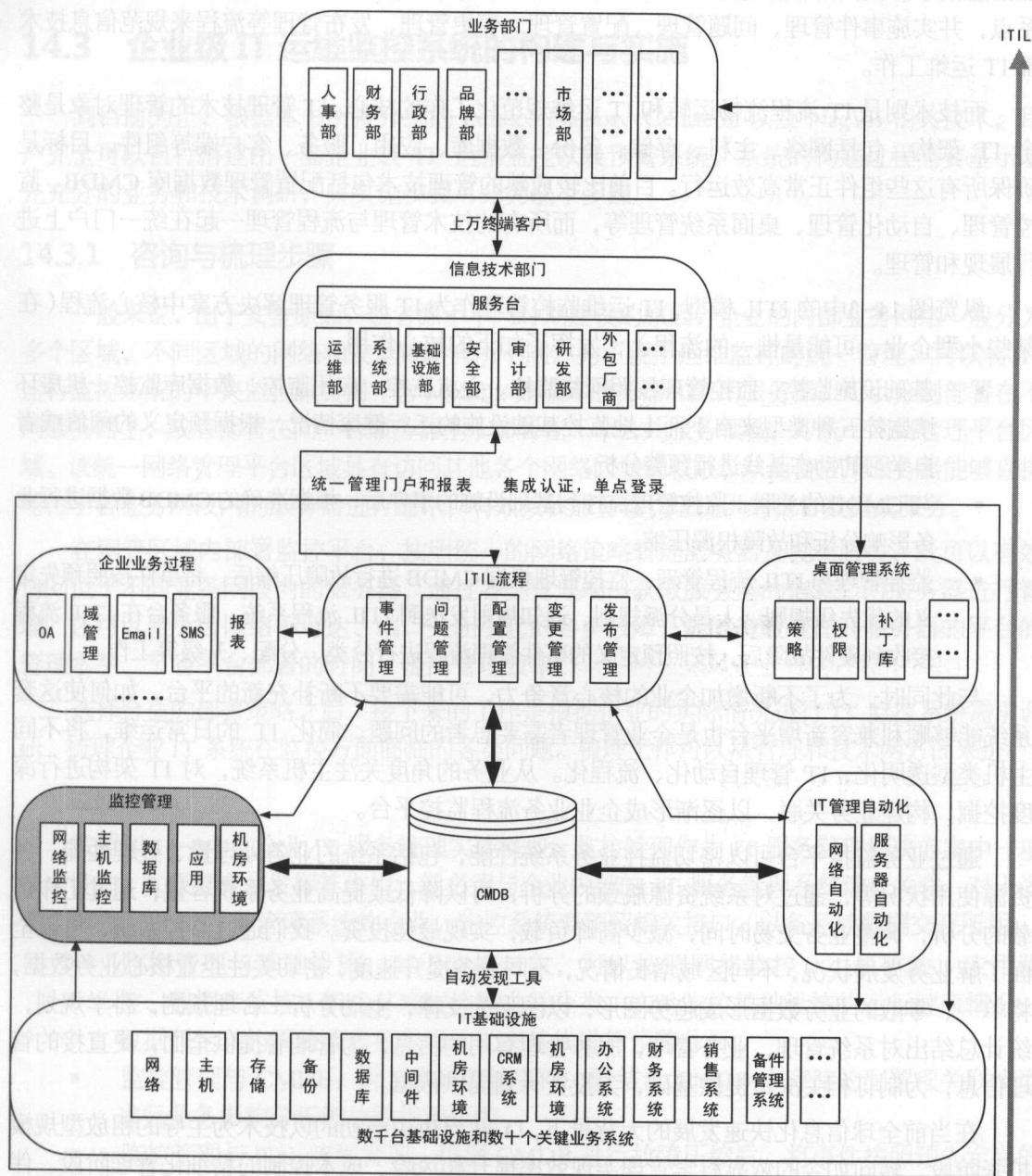


图 14-3 企业 IT 服务管理实施模型

图 14-3 中，人员的概念除了业务部门的用户外，更重要的是包括相关技术人员。IT 部门的技术人员是提供 IT 服务的主体，也是执行模型中相关流程和技术的主体。企业的各项业务部门和负责提供 IT 服务的 IT 部门都将按照基于 ITIL 的最佳实践流程进行有序运转，并且业务部门通过 IT 部门的服务台与信息技术部门进行单点联系。

基于 ITIL 最佳实践的流程是 IT 服务管理整合模型的枢纽。根据当前的管理实践，一般

在信息技术部门内部设立运维监控中心和 IT 服务台，作为业务部门与信息技术部门单一联系点，并实施事件管理、问题管理、配置管理、变更管理、发布管理等流程来规范信息技术部 IT 运维工作。

而技术则是 IT 流程流畅运转和 IT 运维规范化工作的核心。IT 管理技术的管理对象是整个 IT 架构，包括网络、主机、存储、备份、数据库、应用、服务、客户端等组件，目标是确保所有这些组件正常高效运行。目前比较成熟的管理技术包括配置管理数据库 CMDB、监控管理、自动化管理、桌面系统管理等，而所有的技术管理与流程管理一起在统一门户上进行展现和管理。

纵览图 14-3 中的 ITIL 模型，IT 运维监控管理作为 IT 服务管理解决方案中核心流程（在某些小型企业，可能是惟一的流程），发挥了如下的核心作用：

- 基础设施监控：监控管理按照网络监控、主机监控、应用监控、数据库监控、机房环境监控五种类型来自下而上地监控基础设施的运行健康情况，根据预定义的阈值或者自学习的动态基线进行预警分析。
- 与 CMDB 的关联：监控管理监控到基础设施的事件后，根据准确的 CMDB 数据进行业务影响分析和故障根源压缩。
- 监控管理与 ITIL 流程管理：监控管理根据 CMDB 进行故障压缩后，将事件按照预先定义的优先级规则、人员分派规则、通知规则发送到 ITIL 流程系统。服务台在 ITIL 流程接收到硬件故障后，按照预定义的事件管理流程进行分类、分配、升级等工作。

与此同时，为了不断增加企业的核心竞争力，可能需要不断补充新的平台，如何使这套系统能够顺利兼容新增平台也是企业管理者需要思考的问题。简化 IT 的日常运维，将不同主机类型透明化，IT 管理自动化、流程化。从业务的角度关注主机系统，对 IT 架构进行深度挖掘，构建业务关联，以逐渐形成企业业务流程监控平台。

通过业务监控平台可以密切监控业务系统性能，包括系统的业务处理量、处理性能、各资源使用状况等，通过对系统资源瓶颈的分析，可以降低或提高业务系统容量；通过工作负载的分析，调整业务交易时间，减少高峰负载，实现最佳投资。我们通过监控系统，可以全面了解业务发展状况，不同区域增长情况，不同业务提升速度，密切关注企业核心业务数据，将单一、零散的业务数据形成趋势图形，以图形为支撑，主动分析、合理预测，科学规划，统计总结出对系统管理，业务管理，服务管理有用的信息，为管理者提供全面，更直接的管理信息，为制订相关决策提供基础，为投资计划提供依据。

在当前全球信息化快速发展的大背景下，IT 业界也由当初的以技术为主导的粗放型规模扩张阶段，转向如今的依靠科学管理实现效率提升和风险、成本控制的精细化管理阶段。伴随着企业 IT 规模的扩大和 IT 成熟度的提高，各类企业的成本管理、效率管理意识普遍增强。这时，向 IT 管理要效益，要求更高的 IT 服务水平，更强的运营管理能力迫在眉睫。

对 IT 内部运营组织来说，IT 部门在企业的生产、管理环节发挥着重要作用。例如在银行、电信、政府、物流仓储，以及其他生产型企业当中，IT 运维监控管理成为核心业务运作依托的根本手段，也成为企业安全生产、成本控制和效率提升的关键部分。在这种情形下，构建安全高效的企业级 IT 运维监控系统，提升组织内部的 IT 服务水平和建立基于流程的高效率运作机制，可以为企业业务部门提供性价比更高的 IT 服务支持，从而确保业务的安全

高效运转, 缩减运营成本、提高业务盈利能力。

14.3 企业级 IT 运维监控系统的构建与实施

到目前为止, 根据本书所接介绍的 Linux、Nagios、Centreon 以及 NagVis 相关技术, 用户完全可以自行搭建出一套企业级 IT 运维监控以及预警系统。系统的构建过程需要基于事先充分的业务和技术调研, 其实施步骤可分为以下步骤。

14.3.1 咨询与梳理步骤

一般来说, 出于安全原因, 或者源于不同时期建设的原因, 企业的内部业务网络一般分为多个区域, 不同区域的网络和安全策略有所不同。为了便于运维监控的统一管理, 可以将 IT 运行监控系统的中央监控服务器 (含 MySQL 数据库服务器和应用服务器, 可以分别部署在不同服务器上, 或者部署在同一台服务器上) 部署在一个公共服务领域, 即统一网络管理平台区域。该统一网络管理平台区域具有访问其他各个网络区域的权限, 从而使应用服务器能够直接与位于各业务网段内的服务器进行通讯, 有效接收服务器代理提供的报告数据或服务。

在网管区域内部署监控平台, 按照统一的网络策略管理服务器及各类服务, 既可以有效访问位于不同业务网段内的服务器, 通过各类管理端口获取服务器的信息, 对服务器进行管理, 还可以避免因网络不可达、端口关闭引起的各类故障, 能够有效地提升服务器的平台的管理水平、节省平台部署的时间、提高管理效率。

选择合适的部署策略之后, 接下来的工作是根据企业的实际情况进行 IT 监控系统需求调研, 梳理企业 IT 系统在监控方面面临的各类问题。梳理工作可以从管理和技术两方面来进行。

监控管理

按照图 14-3 中的企业 IT 服务管理实施模型, 监控管理作为 IT 服务管理解决方案中一项技术设施, 如果要充分发挥其作用, 就必须与企业内部的 IT 服务管理系统建立关联。对于尚未实施完毕的 IT 服务管理系统的企业, 监控系统要预留相关接口, 以备后续数据交互所用。

- 监控管理与基础设施: 对于监控对象, 可以按照网络监控、主机监控、应用监控、数据库监控、机房环境监控等进行分类, 实时监控基础设施的运行健康情况, 根据预定义的阈值或者自学习的动态基线进行预警分析。
- 监控管理与 CMDB: 对于实时监控事件, 要结合 CMDB 中存储的配置项关联模型进行业务影响分析和故障根源压缩。
- 监控管理与 ITIL 流程管理: 根据 CMDB 进行故障压缩后, 将事件按照预先定义的优先级规则、人员分派规则、通知规则发送到 ITIL 流程系统。服务台在 ITIL 流程接收到硬件故障后, 按照预定义的事件管理流程进行分类、分配、升级等工作。

监控技术

监控技术涉及到基础设施监控、网络设备监控、数据库系统监控、日志监控、应用系统监控等, 详述如下。

- 对于机房基础设施环境进行全天候不间断地运行监控, 包括但不限于温度、湿度、漏水、UPS、空调、视频监控、门禁等。由于机房设施的核心程度和重要性,

应为其制订告警升级策略。上述设施在运行过程中一旦出现实际指标数超过预设的临界数值时,系统报警信息以电子邮件、手机短信、声光报警等方式在第一时间通知给运维人员,要求运维人员根据报警迅速查找原因并进行紧急修复处理。

- 对网络设备进行监控,范围包括各类交换机、路由器、防火墙、带宽管理设备、负载均衡设备、上网行为管理设备等。可以实现设备连接自动搜索、拓扑结构视图呈现、终端跟踪等;能够对 LAN 和 WAN 流量进行监控和排错,能提供网络设备的图像报告、实时流量分析;实现对服务器(操作系统、CPU、内存、磁盘空间等)、应用管理(数据库、表空间、连接数、事务响应等)、中间件、Web 服务器、Email 服务器等的监控和排错,能提供相应的图像报告、实时状态分析;实现网络设备、服务器、应用系统的集中监控、管理。
- 对于各类 7×24 小时不断运行的硬件设备,对于提供了大量应用服务的各类操作系统,通常建议监控如下内容:
 - 服务器状态。
 - CPU: 监控系统 CPU 的占用情况,如 CPU 的利用率等。
 - 硬盘: 磁盘活动时间、磁盘读写速率等指标。
 - 内存: 监控系统内存的状态,内存占用率等。
 - 文件系统: 实时监控文件系统的利用率,如根文件系统、var 文件系统、tmp 文件系统、应用文件系统等。
 - 虚拟内存: 监控虚拟内存的总量、利用率等。
 - 进程: 监控所有重要的进程的启动、停止和状态改变情况。
 - 网络: 监控服务器网络端口的丢包率、利用率、发送速率等指标。
 - 日志: 监控 UNIX 系统的 syslog 日志和 window 的 Event Log。
 - 端口: FTP 端口、HTTP 端口、DNS 端口等。
- 对于企业而言,一旦数据库崩溃或者数据库的性能降低,那么会直接导致依赖于数据库的应用系统运行速度缓慢或者根本无法使用,其最终结果不仅仅是会影响应用系统的使用效率,甚至会造成企业客户和利润的流失。更有甚者,对于某些企业来说,其日常的运营完全依赖于业务系统,那么一旦业务系统所使用的数据库崩溃,就会对企业造成根本性的伤害,或者会影响到企业的正常运营。对于数据库的监控,建议实现以下监控内容:
 - 自动升级警告和警报,并按用户的定义执行纠错例行程序。
 - 为 DBA 提供关键信息以消除瓶颈,或对系统进行调整以获得最高性能。
 - 使 DBA 了解用户所处的性能水平,提高 DBA 有效优化性能和吞吐量的能力。
 - 消除多余的进程,从而提高资源的可用性。
 - 提供重要指标以帮助 DBA 开发更多满足需要的有效容量计划。
 - 帮助 DBA 快速确定并纠正错误根源使数据库保持最高可用性。
 - 日志监控。

一些应用系统输出的应用日志格式由于种种原因,无法直观地查看、分析应用系统运行状况,这也为应用系统的监控带来了难题。这种情况下基于日志进行应用系统监控需要对应用系统的进行大量改造工作。对于重要的、运行多年的应用系统来说,改造工作会给应用系

统的运行带来较大的风险，且投资较大，投入产出比不高，不能称之为一种好的方式。根据经验，对于日志的监控可遵循如下 3 种模式：

- 单行匹配模式：对一行日志进行正则表达式匹配，使用 Shell 或 Perl 的正则表达式即可实现匹配，可参考 Nagios 的著名日志插件 Check_logfiles（http://exchange.nagios.org/directory/Plugins/Log-Files/check_logfiles/details）。
- 段落匹配模式：通过匹配表达式，对连续几行日志作为一条记录进行匹配。将多行日志作为一个整体进行匹配。传统的 Shell/Perl 只能实现单行的正则匹配；为了实现段落匹配的功能，可通过脚本实现扩展匹配算法，可以实现段落匹配。
- 带上下文的单行/段落匹配模式：通过预定义的匹配规则，结合日志内容的上下文进行模式匹配，如图 14-4 所示。

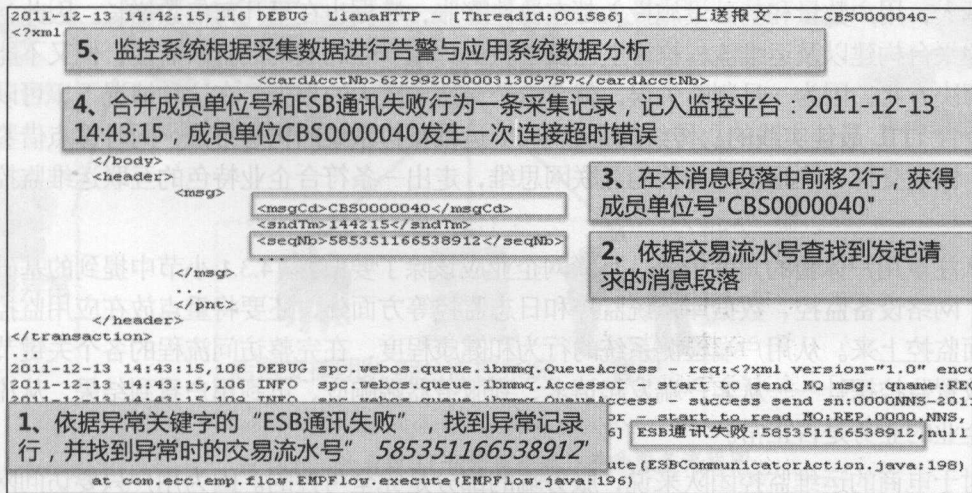


图 14-4 带上下文的单行/段落匹配模式

■ 企业业务服务监控指标的选择与设计

应用系统的功能通常比较复杂，牵扯了较多的业务概念，没有业务人员的配合很难落实，这给项目实施带来了很大的难题。但是从另一方面来说，企业级核心业务及应用系统一般有多个企业内部或者外包团队专家的充分支撑，应用监控解决方案人员在相关资源的支持下能够掌握关键应用系统的关键可用性指标。

一般来说，应用监控解决方案设调研人员应通过规范化、知识化的手段协同应用管理员实现应用系统组成部分的分析，包括基础支撑环境数据、应用依赖环境数据以及系统间关联的指标梳理，并基于这些数据实现拓扑的展示。

如图 14-5 所示的指标模型实现了对应用监控指标的收集。在实施落地层面，需考虑以下几点原则：

- 现有监控方式可落地；
- 对应用系统的性能资源消耗小；
- 投资回报率高；
- 尽量避免应用系统的改造。

系统名称	系统编码	指标模型	监控点名称	资源编码	监控方式	关注重点指标	备注
XX系统	XX	应用服务可用性	登录访问	XX-FW-0001 XX-FW-0002	HTTP模拟	是否成功、响应时长	
		健康度	基础环境				
			应用环境	XX-YY-0001 XX-YY-0002			
			系统关联	XX-XX-0001 XX-XX-0002			
			接口关联	XX-JK-0001 XX-JK-0002			

图 14-5 应用监控指标梳理表格

14.3.2 互联网运维监控实践

相较于规模稳定的传统行业，对于用户数量快速膨胀的互联网公司来说，随着系统体量越来越大，用户数量和基础设施投入都在快速膨胀，数据也在呈几何倍数增长。因此，在运维监控平台构建以及运维流程执行上已很难找到其他企业的成功经验来借鉴，但又不能凭空揣测解决方案，因为一旦判断失误，就会给公司造成巨大的损失。在这种情况下，可以适度采纳符合 ITIL 最佳实践的、传统的企业级 IT 运维监控系统的构建方法，同时重点借鉴敏捷开发、快速交付、注重用户体验的互联网思维，走出一条符合企业特色的互联运维监控实践之路。

从注重用户体验的角度出发，互联网企业应该除了要做好 14.3.1 小节中提到的基础设施监控、网络设备监控、数据库系统监控和日志监控等方面外，还要将重点放在应用监控和业务层面监控上来。从用户端检测系统的行为和健康程度，在完整访问流程的各个关键节点设计并放置监控探针——从客户端发起请求、到服务器端响应、再到用户看到结果，从流程来判断企业的业务是否正常。

对于电商的运维监控团队来说，服务端的部分是完全可控的。因为用户只要访问网站就会留下痕迹，例如从哪来的请求、谁来访问、访问什么 URL、返回的状态是什么、花费多长时间、使用何种终端的哪类浏览器来访问的——是平板电脑还是手机、是 IE 浏览器还是 Firefox 浏览器等等，都可以记录下来进行监控和分析。但是用户的感受是什么样子，例如网站响应速度、图片加载速度、订单提交速度、物流确认速度等等，其实是不可控的，有的用户接入网很差，他的电子购物体验就会很差，这些体验运维团队无法感知到，就必须借助于部署在外围环境的各类监控项来实现。

例如，对于电商网站来讲，创建订单的动作是其核心业务，那么每秒钟创建订单的数量就需要重点监控，如果数量曲线出现突然的波动，例如从每秒 2000 单陡降至每秒 500 单，在此期间，尽管各个服务器的 Load 负载值很低，CPU 也没什么消耗，但是业务性能出现严重下降，这个时候就需要监控系统做出迅速的反应和告警了。

从贴近用户体验的角度来说，可以将涉及到用户体验的相关监控项部署在更为广阔的区域内，例如世界各地、全国各省、各区域，全方位、多角度地对网站服务实时监测，以判断是某个区域的问题，还是整个系统的问题。此类监控数据既可以用来隔离故障区域，判别故障产生的区域，还可以用来消除访问瓶颈，优化用户的访问体验。

14.3.3 提升监控及预警能力

考虑到应用系统不断的升级调整、应用系统自身的复杂性等情况，会造成监控范围覆盖

度不够、监控能力不足等问题，企业应通过研究提供监控能力定期提升服务，对新增资源进行监控、对原有指标的监控阈值、监控策略、监控方式进行调整，保障应用系统监控水平得到及时和稳步提高，详细流程如图 14-6 所示。

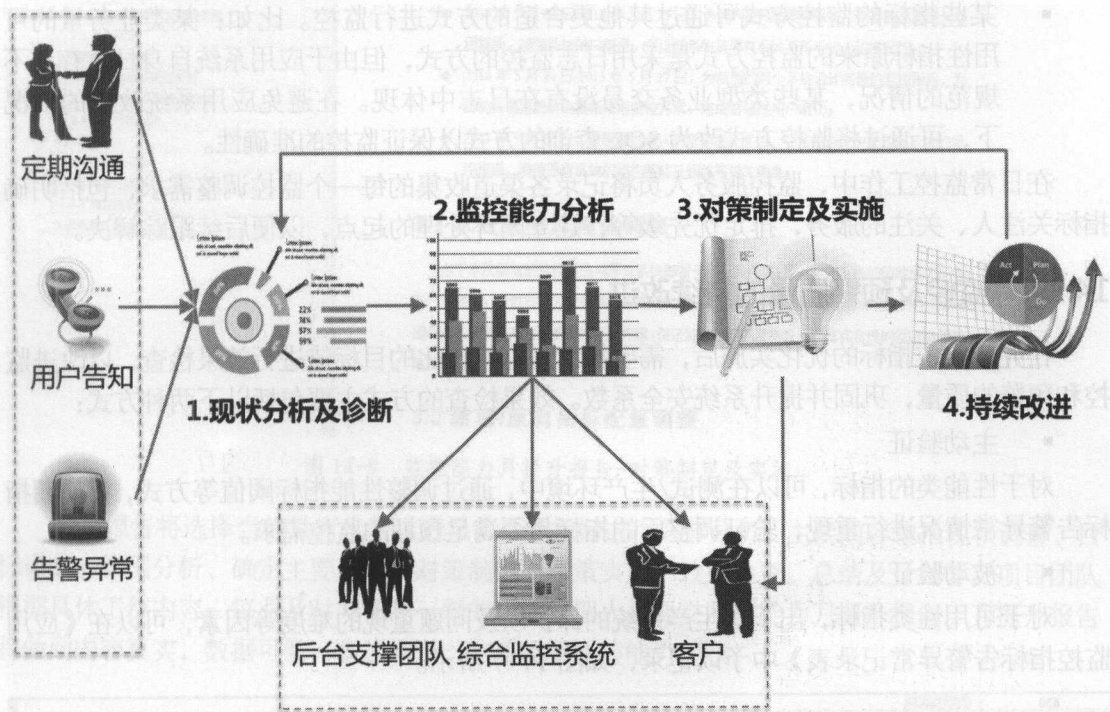


图 14-6 企业级 IT 运维监控能力提升服务流程图

监控能力的提升是一个长期的过程，需要建立一个长效的沟通机制以确保工作的及时性、有效性，监控平台的实施人员与企业相关应用系统管理员通过这种沟通机制有效协作以

确保监控能力的不断提升。作为该沟通机制的起点，监控调整需求的来源包含如下三类：

定期沟通

监控平台的服务人员应定期与系统管理员进行沟通。例如：通过邮件的方式向所有应用系统管理员收集当前应用系统环境变化，将需要纳入监控系统的指标、资源进行调整，从而确保监控指标体系的及时性、有效性，使得监控或告警的结果满足预期或当前需要。

系统管理员告知

由于应用系统优化、升级调整等情况，需要及时进行监控的资源及指标，系统管理员通过变更工单等书面方式主动告知监控服务人员针对调整后的应用系统进行监控支撑。

告警异常

由于应用系统架构较为复杂，所以要定期采取如下措施，以避免可能会出现服务中断但告警未发现的情况：

- 某些阈值/策略设置需要进行有针对性的调优。
- 应用系统运行的不同阶段需要调整。比如：在应用系统上线早期，由于系统运行不稳定，监控阈值设置的相关宽泛，当应用系统运行稳定后，会将阈值设置的趋近

普平■

- 醫平・水空

在日
指标关注

14.3.4

在完
控和预警

-

对于
标告警异

-

对于
监控指标

[illegible]

图 14-7 监控指标告警异常记录表样例

对于
认、验证
段时间内
持续观察
异常指标

最后 编制监控

1 本月监控能力提升服务概述
2 现状分析及诊断
2.1 现状调查
2.2 原因分析
2.2.1 故障原因分类统计
2.2.2 主要原因及影响范围
2.3 目标设定
3 对策制定及实施
3.1 监控策略及阈值调整
3.2 新增/原有指标配置调整
3.3 二次开发支持
4 效果检查
4.1 性能类指标
4.2 可用性类指标
4.3 其它指标
5 总结及计划

3 对策制定及实施

根据目标的设定，我们将此次监控指标调整工作严格控制在 5 个工作日内，具体工作的进度安排如下：

- ◆ 2013 年 5 月 22 日-2013 年 5 月 23 日，协调内控风险测评系统厂商实施人员进入项目现场，调整系统端口配置，保证新系统及原有系统互不干扰，正常运行；
- ◆ 2013 年 5 月 24 日-2013 年 5 月 27 日，为新增的两个系统设计完善的监控指标，为货物贸易监测系统重新规划监控方案，完成后通过用户确认；
- ◆ 2013 年 5 月 28 日，按照通过确认的指标设计文档，在监控系统中配置新系统的监控指标，调整原有系统的监控策略及阈值等相关信息。

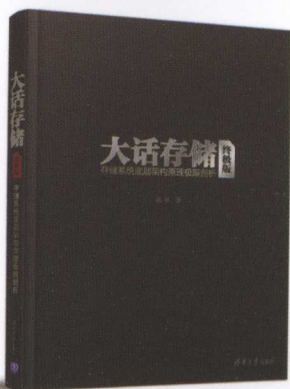
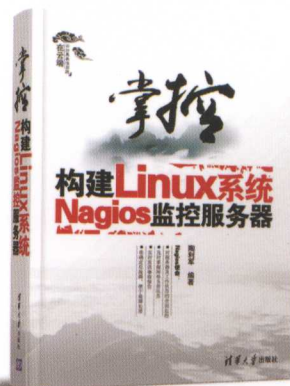
3.1 监控策略及阈值调整

由于本次新增的监控系统与原有系统部署于同一服务器，原有应用系统的负载情况将发生一定变化。通过对相关业务系统的应用状况进行调查，我们重新对监控阈值进行了分析，提供并实施了更为合理的监控优化方案，保证监控指标以及告警能够真实合理的反应当前整体系统运行的状态信息。

3.2 新增/原有指标配置调整

图 14-8 监控能力月提升报告-对策制定及实施

提升报告将选择当前具有代表性的异常告警指标进行具体分析，主要内容包括现状调查、目标设定、原因分析、确定主要原因、对策制定、对策实施、效果检查、总结及计划等。项目团队根据具体工作内容，每月及时向更高一级的运维管理人员提交监控能力月提升报告。月提升报告将做到内容真实，数据可靠，表述清晰，准确反映监控能力现状。



海量运维监控 系统规划与部署

(增补Linux+Nagios+Centreon+NagVis版)



用“工匠精神”对待IT运维监控工作

- ◎ 不仅要让IT运维监控平台工作，更要注重运维监控规范的建立
- ◎ 不仅满足短期的运维监控目标，更要构建符合自身特点的运维监控产品体系
- ◎ 不仅可以响应监控需求的变化，更要形成自己的运维方法论
- ◎ 不仅要有个体的钻研，更要参与互联网社区并乐于分享知识

实际操作是提升技能的好方法，本书为您分享有深度、有思想、有价值的运维监控知识，助您不断重复实践，从学徒走向大师。

清华大学出版社数字出版网站

WQBook 书文

www.wqbook.com

【上架提示】

IT运维/监控/Linux

ISBN 978-7-302-40953-3



9 787302 409533 >

定价：59.00元